
OmniZoom: A Universal Plug-and-Play Paradigm for Cross-Device Smooth Zoom Interpolation

Xiaoan Zhu*

Zhejiang University
xiaoanzhu@zju.edu.cn

Yue Zhao*

Huawei Noah's Ark Lab
zhaoyue53@huawei.com

Tianyang Hu

Zhejiang University
huty@zju.edu.cn

Jiaming Guo

Huawei Noah's Ark Lab
guojiaming5@huawei.com

Yulan Zeng

Sungkyunkwan University
zengyulan@skku.edu

Renjing Pei

Huawei Noah's Ark Lab
peirenjing@huawei.com

Fenglong Song

Huawei Noah's Ark Lab
songfenglong@huawei.com

Huajun Feng[†]

Zhejiang University
fenghj@zju.edu.cn

Abstract

Dual-camera smartphones suffer from geometric and photometric inconsistencies during zoom transitions, primarily due to disparities in intrinsic/extrinsic parameters and divergent image processing pipelines between the two cameras. Existing interpolation methods struggle to effectively address this issue, constrained by the lack of ground-truth datasets and motion ambiguity in dynamic scenarios. To overcome these challenges, we propose OmniZoom, a universal plug-and-play paradigm for cross-device smooth zoom interpolation. Specifically, we present a novel cross-device virtual data generation method utilizing 3D Gaussian Splatting. This method tackles data scarcity by decoupling geometric features via spatial transition modeling and correcting photometric variations with dynamic color adaptation. It is further enhanced by cross-domain consistency learning for device-agnostic semantic alignment. Additionally, we introduce a plug-and-play 3D Trajectory Progress Ratio (3D-TPR) framework that surmounts 2D spatial limitations. As components of our framework, a texture-focus strategy is introduced for high-frequency detail preservation, incorporating mask penalty constraints to suppress interpolation artifacts. Our pipeline exhibits broad compatibility with diverse interpolation methods and achieves good performance across multiple public benchmarks. Real-world evaluations on various smartphone platforms also reveal significant quality improvements after finetuning on our synthetic data, which underscores the robustness and practical effectiveness of our approach for cross-device zoom applications.

1 Introduction

In recent years, dual-camera systems [8, 9, 13] have become increasingly prevalent in modern smartphones, offering advanced imaging capabilities. Among these functionalities, zooming [2, 58] plays a crucial role by bridging the wide-angle and main cameras, allowing users to perceive changes in subject distance and scale without physical movements. However, performing optical zoom [10, 28]

*Equal contribution.

[†]Corresponding author.

introduces significant geometric misalignment and photometric inconsistencies between images. Geometric misalignment arises from disparities in the intrinsic and extrinsic parameters of the dual-camera system [46, 57]. In contrast, photometric inconsistencies are caused by algorithmic divergences (e.g., white balance, color correction, and gamma correction) in Image Signal Processing pipelines (ISP) [36, 26]. These differences lead to parallax, color shifts, and exposure mismatches between the paired images, severely affecting the smoothness and realism of zoom transitions.

Notably, the transition between fixed-focus dual-camera views can be naturally formulated as an intermediate frame synthesis task. Frame interpolation (FI) [17, 25, 27, 53, 60, 47, 3] is a promising technique in this context, as it aims to synthesize temporally consistent intermediate frames between given inputs. In this work, we define **zoom interpolation (ZI)** as a specialized sub-task of FI, where the objective is to synthesize an intermediate frame that is both geometrically and photometrically consistent, given two zoomed images captured from a dual-camera system. While ZI shares the core goal of FI in generating intermediate content, it also presents several unique challenges that hinder the direct application of conventional FI methods.

First, unlike FI tasks with dense temporal supervision, ZI lacks suitable ground-truth intermediate frames. This is because deploying a continuous-zoom [2, 58] capture setup across dual-camera hardware is impractical in real-world mobile scenarios, making it difficult to train or finetune models in a supervised manner. Second, ZI suffers from severe motion ambiguity. Traditional FI approaches rely on fixed timestep maps [29, 3] to model pixel displacement, assuming smooth linear motion. However, zoom transitions often introduce complex, nonlinear object motion and parallax effects. These distort assumptions in flow estimation, leading to motion blur and structural artifacts.

To address the unique challenges of ZI, we propose OmniZoom, a universal plug-and-play paradigm for cross-device smooth zoom interpolation. This paradigm comprises two key components specifically designed for this task: **(1) Generation of cross-device smooth zoom virtual dataset.** To overcome the lack of ground-truth intermediate frames in ZI, we build upon ZoomGS [50] and adopt 3D Gaussian Splatting (3DGS) [23] to synthesize high-quality data. While ZoomGS is limited to single-device scenarios, our method further resolves cross-device geometric and photometric inconsistencies. We enhance this process with spatial transition modeling and dynamic color adaptation to decouple geometric and photometric differences between devices. Furthermore, a cross-domain consistency learning scheme ensures the generated virtual dataset maintains semantic alignment and photometric fidelity across different camera platforms. **(2) 3D-TPR framework.** To tackle spatial deficiencies and velocity ambiguity during interpolation, we propose a plug-and-play 3D-TPR module that encodes 3D motion-aware correspondence across frames. In addition, we incorporate a texture-focus strategy to preserve high-frequency details and a mask uncertainty penalty to suppress ambiguous predictions. These components work jointly to improve perceptual sharpness and mitigate artifacts during zoom synthesis.

Our pipeline is specifically designed for ZI and exhibits strong plug-and-play adaptability across a variety of FI backbones and mobile devices. By integrating the 3D-TPR framework into various existing architectures, we observe remarkable fidelity improvements and superior subjective quality on multiple public benchmarks. To further validate its real-world applicability, we construct test datasets captured from various smartphones. Finetuning FI models on our ZI dataset leads to substantial quality gains across all models on the real-world benchmark. Results demonstrate the generalizability and robustness of our approach in addressing zoom-related challenges across diverse FI models and mobile devices, providing a unified solution for achieving smooth and perceptually coherent zoom transitions in dual-camera systems.

In conclusion, our key contributions are as follows:

- We propose a novel pipeline for constructing a cross-device ZI dataset, which models geometric and photometric inconsistencies across devices. The resulting dataset enables high-quality, device-agnostic supervision, filling the gap in ground-truth data for ZI task.
- We introduce a novel ZI framework, 3D-TPR, which enhances spatial modeling via disparity-aware encoding, complemented by a texture-focus strategy and a mask penalty for better perceptual quality and robustness.
- We propose OmniZoom, a universal plug-and-play paradigm for cross-device smooth zoom interpolation that seamlessly integrates with FI networks. It consistently delivers high-quality ZI results across diverse FI models and smartphone devices in real-world scenarios.

2 Related works

2.1 Frame interpolation methods

Frame Interpolation plays a critical role in high-resolution video synthesis, slow-motion rendering, editing, and AR/VR applications [30, 33, 16]. Recent advances have explored temporal mapping strategies to improve interpolation quality. For instance, Interpany[60] replaces fixed timestep maps with 2D flow-based distance-aware index. However, its reliance on 2D optical flow limits robustness under complex 3D motion and large depth variation. We propose a 3D geometric-aware encoding mechanism that captures spatial coherence more effectively. Unlike methods such as [17] that suppress artifacts utilizing deep hierarchical features, our design handles mask uncertainty without additional architectural complexity. Despite advancements of recent learning-based FI models [3, 53, 4, 27, 19, 17, 34, 25], restoring fine textures under large motion remains challenging. We tackle this via a texture-focus training strategy that reweights reconstruction loss based on gradient cues, offering a lightweight alternative to traditional perceptual losses [20] or multi-scale losses [44].

2.2 3D reconstruction

3D reconstruction has evolved from classical geometry-based methods such as Structure-from-Motion (SfM) [38] and Multi-View Stereo (MVS) [14], which enable scene recovery from unstructured images to deep learning approaches that enhance robustness and generalization [12, 61, 43]. Neural Radiance Fields (NeRF) [31] and its variants (e.g., Mip-NeRF 360 [6]) achieve high-fidelity view synthesis but at high computational cost. Recently, 3D Gaussian Splatting (3DGS) [23] has emerged as an efficient alternative with explicit point-based representation and real-time rendering capability. ZoomGS [50] leverages 3DGS to create virtual cameras between dual-camera pairs for interpolation supervision, but its device-specific calibration limits generalization. In this work, we extend 3DGS and ZoomGS into a universal, cross-device virtual data generation pipeline, jointly addressing geometric and photometric inconsistencies to support consistent and smooth zoom synthesis across diverse mobile platforms.

3 Method

In this section, we first present a pipeline based on 3DGS [23] for constructing a cross-device smooth zoom virtual dataset. This dataset enables the generation of high-quality virtual images that are not limited to a specific camera module. Building upon this foundation, we propose the 3D-TPR framework, which enhances temporal consistency, preserves fine details, and improves spatial reliability during interpolation. The overall pipeline is illustrated in Figure 1(a).

3.1 Generation of cross-device smooth zoom virtual dataset

ZoomGS [50] utilizes 3DGS [23] to synthesize virtual 3D camera models $\mathcal{M}_v$ between the main camera $\mathcal{M}_m$ and wide-angle camera $\mathcal{M}_w$, facilitating the generation of pseudo-supervised training images $\tilde{I}$ via:

$$\tilde{I} = \mathcal{R}(\mathcal{M}, \mathcal{K}, \mathcal{P}), \quad (1)$$

where $\mathcal{R}$ denotes the 3DGS rendering process [23], and $\mathcal{K}, \mathcal{P}$ represent the intrinsic and extrinsic parameters of the corresponding cameras. However, the dataset generated by ZoomGS is device-dependent and lacks generalizability across different smartphone platforms. To overcome this limitation, we extend the ZoomGS pipeline to construct cross-device ZI datasets applicable to arbitrary smartphone configurations. As shown in Figure 1(b), we introduce a spatial transition modeling module and a color adaptation strategy that decouple the geometric and photometric characteristics of individual camera modules. Furthermore, we propose a cross-domain consistency learning approach to enforce semantic alignment and visual coherence across devices and environments.

Spatial transition modeling. For each scene, we capture N zoom image pairs comprising wide-angle images $\{I_w^n\}_{n=1}^N$ and main images $\{I_m^n\}_{n=1}^N$. These images are then processed using COLMAP [38] to calibrate their intrinsic parameters $\{\mathcal{K}_w^n\}_{n=1}^N$, $\{\mathcal{K}_m^n\}_{n=1}^N$ and extrinsic parameters $\{\mathcal{P}_w^n\}_{n=1}^N$, $\{\mathcal{P}_m^n\}_{n=1}^N$. For each pair, we generate M camera poses, including the two original cameras and $M-2$ interpolated virtual views in between. To achieve smooth spatial transitions, we interpolate both intrinsic and extrinsic parameters between the two real cameras. For the intrinsic parameters $\mathcal{K}_{v_i}$, we perform linear interpolations [15]:

$$\mathcal{K}_{v_i} = (1 - \frac{i-1}{M-1}) \odot \mathcal{K}_w + \frac{i-1}{M-1} \odot \mathcal{K}_m, \quad (2)$$

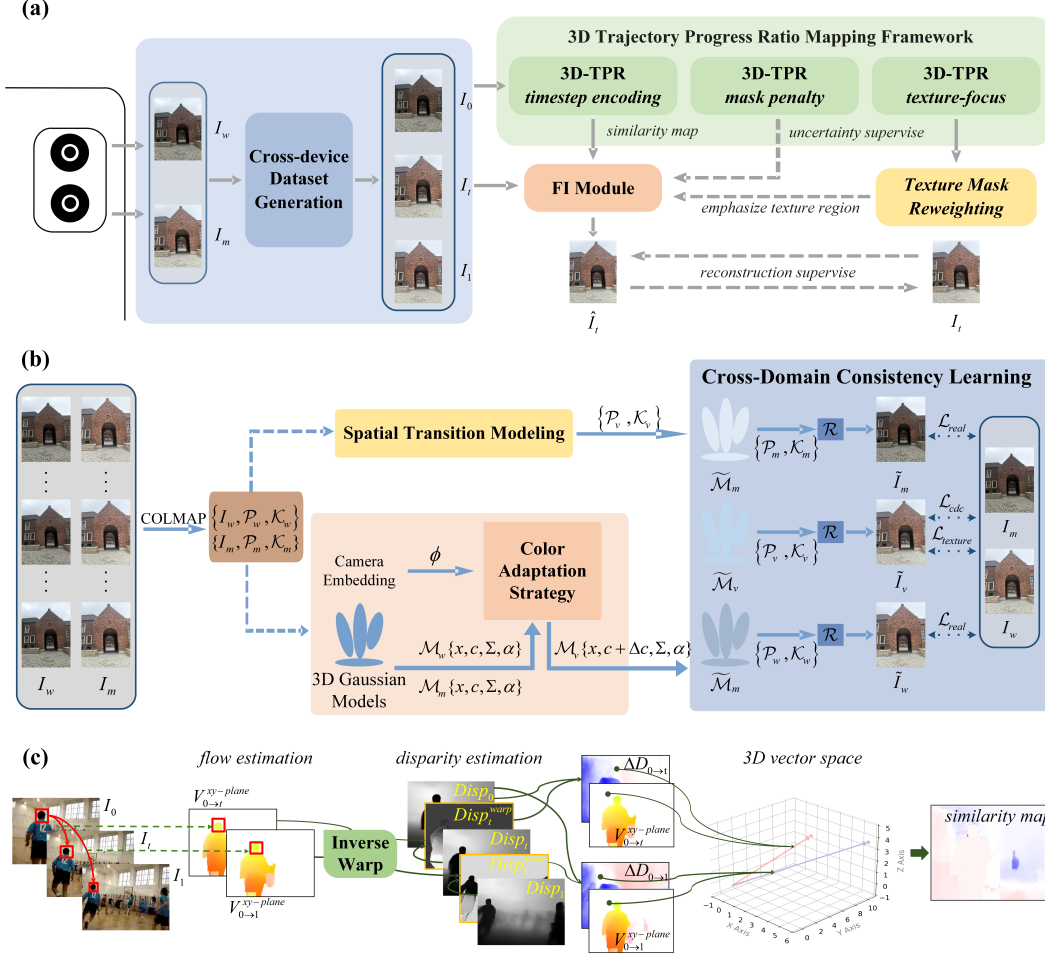


Figure 1: Overview of our proposed OmniZoom. (a) Plug-and-play integration pipeline for cross-device ZI. (b) Dual-camera ZI dataset generation via spatial-color calibration and cross-domain optimization. (c) Construction of similarity map in the 3D-TPR encoding module.

while the extrinsic parameters $\mathcal{P}_{v_i}$ are interpolated in Euclidean space [21, 5]:

$$\mathcal{P}_{v_i} = \mathcal{P}_w \odot \exp \left(\frac{i-1}{M-1} \odot \log \left(\frac{\mathcal{P}_m}{\mathcal{P}_w} \right) \right), \quad (3)$$

where $i \in 1, 2, \dots, M-1$. This spatial transition module enables continuous and geometrically coherent virtual viewpoints by interpolating between COLMAP-estimated camera poses. It models zoom motion purely from relative transformations, without relying on any device-specific parameters.

Color adaptation strategy. To handle photometric inconsistencies across heterogeneous devices, we introduce a dynamic color adaptation mechanism that operates at the level of individual 3D Gaussians. Specifically, we represent each camera as a 3DGS model $\mathcal{M}$ composed of a set of 3D Gaussians. Each Gaussian is parameterized by its center position x , color c , anisotropic covariance Σ (indicating its spatial extent), and opacity α [23]. To account for photometric shifts between devices, we assign each camera a learnable embedding ϕ (e.g., $\phi_w = 1$ for wide-angle and $\phi_m = 0$ for main camera), and interpolate the virtual camera embedding ϕ_{v_i} following Equation 2. A lightweight neural network, ColorNet $\mathcal{C}$, then predicts a per-Gaussian color offset based on both the geometric and appearance attributes:

$$\Delta c = \mathcal{C}(\phi, x, c, \Sigma, \alpha). \quad (4)$$

This offset is applied to each Gaussian, resulting in a view-specific virtual model $\tilde{\mathcal{M}}_{v_i}$ with updated color values $\{x, c + \Delta c, \Sigma, \alpha\}$. This learned photometric correction is continuous and device-agnostic, enabling robust color alignment across diverse camera modules. Further architectural details and ablation studies of ColorNet are provided in the Appendix A.1.

Cross-domain consistency learning approach. To ensure semantic and visual consistency across devices, we introduce a cross-domain consistency learning strategy. During joint optimization of the color adaptation network $\mathcal{C}$, the rendered images $\tilde{I}_w$ and $\tilde{I}_m$ are supervised by their corresponding ground truth image I_w and I_m via a reconstruction loss:

$$\mathcal{L}_{real} = (1 - \lambda)\mathcal{L}_1(\tilde{I}_{real}, I_{real}) + \lambda\mathcal{L}_{SSIM}(\tilde{I}_{real}, I_{real}), \quad (5)$$

where $real \in \{m, w\}$, and $\mathcal{L}_1$, $\mathcal{L}_{SSIM}$ denote the ℓ_1 loss and SSIM loss [49], respectively. We follow the default hyperparameters from 3DGS and set $\lambda = 0.2$. However, relying solely on low-level pixel reconstruction is insufficient to ensure perceptual consistency across domains. To enhance semantic structure and texture realism, we introduce two additional feature-space constraints. We propose a semantic consistency loss $\mathcal{L}_{cdc}$ based on a pre-trained lightweight feature extractor $\mathcal{F}$ [37]:

$$\mathcal{L}_{cdc} = \|\mathcal{F}(\tilde{I}_{v_i}) - \mathcal{F}(I_{real})\|_1, \quad (6)$$

Additionally, to preserve texture consistency across domains, we incorporate a texture consistency loss $\mathcal{L}_{texture}$, computed using a VGG-based texture encoder $\mathcal{T}_{texture}$ [39]:

$$\mathcal{L}_{texture} = \|\mathcal{T}_{texture}(\tilde{I}_{v_i}) - \mathcal{T}_{texture}(I_{real})\|_2 \quad (7)$$

The final objective for domain-consistent virtual image generation is given by:

$$\mathcal{L}_{total} = \mathcal{L}_{real} + \mathcal{L}_{cdc} + \mathcal{L}_{texture} \quad (8)$$

3.2 3D-TPR framework

Traditional FI models represent the interpolation time t using a fixed timestep map that is concatenated with the input frames $\{I_0, I_1\}$. However, this static encoding lacks adaptability to scene content and motion dynamics. As illustrated in Figure 1(a), our 3D-TPR framework replaces the fixed timestep map with a pixel-wise similarity map computed using 3D-TPR encoding, which captures 3D geometric relationships between the input views. In addition, we introduce a texture-focus strategy and a mask-based uncertainty penalty to provide auxiliary supervision.

3D-TPR encoding. As shown in Figure 1(c), $\{I_0, I_1, I_t\}$ serves as the input to the core 3D-TPR encoding module. Following InterpAny [60], we use RAFT [42] to estimate 2D pixel-wise optical flows $\{V_{0 \rightarrow t}^{xy-plane}, V_{0 \rightarrow 1}^{xy-plane}\}$. We compute the corresponding backward flows $\{V_{t \rightarrow 0}^{xy-plane}, V_{1 \rightarrow 0}^{xy-plane}\}$ using an inverse warp module $W^{-1}(\cdot)$:

$$V_{t \rightarrow 0}^{xy-plane} = W^{-1}(V_{0 \rightarrow t}^{xy-plane}), \quad V_{1 \rightarrow 0}^{xy-plane} = W^{-1}(V_{0 \rightarrow 1}^{xy-plane}). \quad (9)$$

Concurrently, we employ an efficient network Lite-Mono [54] to predict monocular disparity maps $\{Disp_0, Disp_t, Disp_1\}$ from the input frames $\{I_0, I_t, I_1\}$. The predicted disparities $\{Disp_t, Disp_1\}$ are then warped to the reference view I_0 using the backward flows via a simple warp operation $W(\cdot)$, yielding the warped disparities $\{Disp_t^{warp}, Disp_1^{warp}\}$:

$$Disp_t^{warp} = W(Disp_t, V_{t \rightarrow 0}^{xy-plane}), \quad Disp_1^{warp} = W(Disp_1, V_{1 \rightarrow 0}^{xy-plane}). \quad (10)$$

We then compute the disparity residuals $\{\Delta D_{0 \rightarrow t}, \Delta D_{0 \rightarrow 1}\}$ by measuring the differences between the warped disparity maps $\{Disp_t^{warp}, Disp_1^{warp}\}$ and the reference disparity map $Disp_0$:

$$\Delta D_{0 \rightarrow t} = \|Disp_t^{warp} - Disp_0\|_\psi, \quad \Delta D_{0 \rightarrow 1} = \|Disp_1^{warp} - Disp_0\|_\psi, \quad (11)$$

where $\|\cdot\|_\psi$ denotes a robust error norm. These components jointly yield the disparity variation vector $Var^z = \Delta D_{0 \rightarrow t} / \Delta D_{0 \rightarrow 1}$, which reflects the disparity differential ratio of the target frame I_t . We hypothesize that this progress-ratio-like scalar can serve as a learned timestep prior, replacing fixed interpolation index and enabling better temporal alignment under non-uniform motion. Consequently, we fuse the disparity residual $\Delta D_{0 \rightarrow t}$ and the planar motion vector $V_{0 \rightarrow t}^{xy-plane}$ into a unified 3D motion field $\Delta V_{0 \rightarrow t}^{xyz}$ via a cross-domain aggregation module $A(\cdot)$. This multimodal fusion is symmetrically extended to the temporal span $0 \rightarrow 1$. Each spatial location $(i, j, k) \in \mathbb{R}^3$ in 3D motion fields is defined as:

$$(\Delta V_{0 \rightarrow t}^{xyz})^{i,j,k} = A(\Delta D_{0 \rightarrow t}, V_{0 \rightarrow t}^{xy-plane}), \quad (\Delta V_{0 \rightarrow 1}^{xyz})^{i,j,k} = A(\Delta D_{0 \rightarrow 1}, V_{0 \rightarrow 1}^{xy-plane}). \quad (12)$$

The trajectory progress similarity at (i, j, k) is then computed via cosine similarity:

$$Sim(i, j, k) = \frac{\Delta V_{0 \rightarrow t}^{xyz}(i, j, k) \cdot \Delta V_{0 \rightarrow 1}^{xyz}(i, j, k)}{\|\Delta V_{0 \rightarrow t}^{xyz}(i, j, k)\| \cdot \|\Delta V_{0 \rightarrow 1}^{xyz}(i, j, k)\|}. \quad (13)$$

This formulation effectively decouples perspective projection effects from true 3D motion, addressing a key limitation of conventional 2D optical flow methods while preserving temporal coherence across interpolated frames.

Gradient-coupled texture-focus training strategy. To mitigate the common tendency of FI networks to oversmooth high-frequency content [52], we introduce a texture-focus reweighting scheme. We identify texture regions in the ground-truth frame I_t using a multi-scale, kernel-based detector $\mathcal{T}$:

$$M = \mathcal{T}(I_t) \in \{0, 1\}^{H \times W}, \quad (14)$$

where H and W denote the height and width of I_t . To enhance texture mask robustness via conventional detection [59], we adopt the Average Local Pixel Difference (ALPD) metric. Specifically, local and global texture metrics are computed as:

$$v_{\text{ALPD}} = \frac{1}{P} \sum_{i=1}^P |g_i - \bar{g}|, \quad v_{\text{global}} = \frac{w}{N_p} \sum_{j=1}^{N_p} |g_j - \bar{g}_{\text{global}}|, \quad (15)$$

where g_i is the intensity of the i -th local pixel ($i = 1, \dots, P$), $\bar{g}$ is their mean, and N_p is the pixel count in I_t . Scalar w serves as a sensitivity coefficient. A pixel is classified as textured if $v_{\text{ALPD}} > v_{\text{global}}$. We then define a reweighted loss using this texture mask:

$$\mathcal{L}_{\text{texture-focus}} = \frac{1}{H \times W} \sum_{x,y} w(x,y) \cdot \rho(\hat{I}_t(x,y) - I_t(x,y)), \quad w(x,y) = 1 + \lambda M(x,y), \quad (16)$$

where λ controls the emphasis on textured areas and $\rho(\cdot)$ denotes the Charbonnier loss. $\hat{I}_t(x,y)$ and $I_t(x,y)$ indicate the predicted and ground-truth frames, respectively. This formulation encourages more accurate reconstruction of real high-frequency details.

Mask uncertainty penalty limitations. Modern FI systems typically synthesize intermediate frames by fusing two consecutive inputs I_0 and I_1 using a dual-branch formulation:

$$\mathbf{F}^{(i)} = \mathbf{F}_0^{(i)} \odot \mathbf{M}^{(i)} + \mathbf{F}_1^{(i)} \odot (1 - \mathbf{M}^{(i)}), \quad (17)$$

where $\mathbf{F}_0^{(i)}$ and $\mathbf{F}_1^{(i)}$ is the i -th scale features of the two input frames, $\odot$ is element-wise multiplication, and $\mathbf{M}^{(i)} \in [0, 1]^{H_i \times W_i}$ is a learnable mask indicating the per-pixel dominance of each source. We posit that under ideal conditions where $\mathbf{M}^{(i)}$ should converge to binary values (0 or 1) for confident fusion, $\mathbf{M}^{(i)}$ around 0.5 often leads to ghosting or blur in synthetic frames. To address this, we introduce an uncertainty-aware regularization term applied to each mask level during training:

$$\mathcal{L}_{\text{mask}}^{(i)} = \lambda_i \cdot \frac{1}{H \times W} \sum_p \left[4 \cdot \mathbf{M}_p^{(i)} (1 - \mathbf{M}_p^{(i)}) \right]^\gamma, \quad 0 < \lambda_1 < \lambda_2 < \dots < \lambda_S, \quad (18)$$

where p traverses all pixel positions, $\gamma \in (0, 1]$ controls the penalty growth rate, and $\lambda_i (i = 1, 2, \dots, S)$ is i -th scale-specific weight. Lower-resolution scales are allowed to retain higher uncertainty to better handle large motion and occlusion, while higher-resolution scales are encouraged to produce more confident, near-binary decisions by assigning larger λ_i . Additional formulation details are provided in Appendix A.4.

4 Experiments

4.1 Implementation

Datasets for virtual camera generation. We capture zoom image pairs from HuaweiPura70Ultra and RedmiK50Ultra [50] across 132 real-world scenes (79 indoor, 53 outdoor), covering diverse environments and illumination conditions. Based on Section 3.1, we generate 205 virtual sequences (16 frames each) for our ZI dataset. For ZI dataset construction details and representative samples, please refer to Appendix A.2 and A.3. We compare our ZI dataset with ZoomGS and 3DGS in terms of supervision quality and generalization ability for real-world ZI. Visual comparisons and finetune results are provided in Appendix A.5.

Similarity map encoding computation. During **training**, $t = \text{Sim}(i, j, k), \forall (i, j, k)$. $\text{Sim}(i, j, k)$ is derived from ground-truth frames using RAFT and Lite-Mono (see Eq 13). During **inference**, the



Figure 2: Sample triplets from our ZI dataset, each showing the wide-angle I_w , the synthetic intermediate frame I_v , and the main camera image I_m , for two devices: Huawei and Redmi.

ground-truth frame is not accessible, and the exact similarity map cannot be computed. Following the strategy used in prior work such as InterAny [60], it is sufficient to provide a $t = \text{fixed} - \text{timestep map}$ in the same manner as conventional time-indexing methods. The semantics of this indexing map have shifted from an uncertain timestep map to a more deterministic motion hint. Physically, this encourages the model to move each object at constant speeds along their trajectories. In practice, this constant-speed assumption serves as a valid approximation for smooth zoom scenarios commonly found in mobile photography.

Training configuration. For virtual camera generation, we adopt the Adam optimizer [24] in a two-stage scheme. We first train the Gaussian model $\mathcal{M}_w$ for 5k iterations using the learning rate from 3DGS [23]. Then, joint optimization is performed for 30k iterations on both $\mathcal{M}_w$ and the ColorNet, with the learning rate decaying from $1e-3$ to $1e-6$ after 20k steps. Our 3D-TPR framework is integrated into RIFE [17], IFRNet [25], EMAVFI [53], and AMT [27]. Each model is initially trained on the Vimeo-90K [52] septuplet dataset (91,701 sequences at 448×256 resolution) using its default hyperparameters. We then finetune them on our ZI dataset. All experiments are implemented by PyTorch [35] on hardware devices equipped with 40 GB of memory.

Evaluation configuration. We evaluate performance and generalization on two types of datasets. For benchmarks with ground-truth, including Vimeo90K, Inter4K [41], and UCF101 [40], we use full-reference image quality metrics: PSNR [18], SSIM [49], and LPIPS [55]. For real-world evaluation, we collect zoom image pairs from four smartphone platforms: HuaweiPura70Ultra, iPhone16ProMax, OPPOFindX8Ultra, and RedmiK50Ultra[50]. Since ground truth is unavailable, we adopt no-reference metrics including NIQE [32], PI [1], CLIP-IQA [45], MUSIQ [22] and FLOLPIPS [11]. For all comparisons, the 2D baseline refers to the distance indexing paradigm [60]. For real-world deployment without geometric priors, the model uses a uniform map as motion guidance when inference. Real-world dataset details are provided in the Appendix A.7.

4.2 Ablation studies

To assess the effectiveness and compatibility of each component in our 3D-TPR framework, we perform a controlled ablation study on EMA-VFI in Table 1. The 2D framework [60] serves as the baseline. We evaluate four configurations: (1) **3D-t-m**, which applies only the 3D-TPR encoding; (2) **3D-m**, which integrates the texture-focus strategy; (3) **3D-t**, which incorporates the mask penalty constraint; and (4) **3D**, the full model with all components enabled.

Although improvements on pixel-centric metrics (PSNR and SSIM) appear modest, we argue that in most practical ZI scenarios, the objective is not to produce pixel-aligned frames, but to **synthesize perceptually plausible intermediate frames**. Pixel-centric metrics are relatively insensitive to common ZI artifacts such as blur and ghosting [56], which our design explicitly targets by **resolving velocity ambiguity**. As such, pixel-centric metrics are less informative for perceptual assessment. While the improvements on pixel-centric metrics (PSNR and SSIM) appear numerically modest, we emphasize that in practical Zoom Interpolation (ZI) scenarios, the objective is not to reproduce pixel-aligned frames but to **synthesize perceptually plausible intermediate frames**. Moreover, pixel-centric measures are inherently **insensitive to blur and ghosting artifacts**—the dominant

Table 1: Ablation study of the 3D-TPR framework. **Bold** values indicate the best results. Each component contributes incremental gains, and the full model yields the best performance.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PI $\downarrow$	MUSIQ $\uparrow$	FLOLPIPS $\downarrow$
2D	24.73	0.851	0.081	5.071	55.968	0.128
3D-t-m	24.78	0.852	0.081	5.041	56.615	0.119
3D-m	24.74	0.852	0.084	5.022	57.362	0.113
3D-t	24.80	0.853	0.081	4.934	56.946	0.116
3D (full)	24.86	0.853	0.080	4.802	58.725	0.091

Table 2: Comparing the impact of optical flow on Vimeo90K.

Benchmarks	Metrics	IFRNet				AMT			
		1x	2x	4x	8x	1x	2x	4x	8x
Vimeo90k	PSNR $\uparrow$	27.21	27.21	26.01	25.01	27.25	27.22	26.01	24.94
	SSIM $\uparrow$	0.901	0.901	0.867	0.847	0.902	0.902	0.849	0.832
	LPIPS $\downarrow$	0.070	0.074	0.085	0.094	0.083	0.084	0.106	0.127

degradations caused by velocity ambiguity, which our framework is explicitly designed to mitigate. Consequently, these metrics are less indicative of perceptual fidelity.

In contrast, perceptual metrics such as PI, MUSIQ, and FLOLPIPS reveal consistent and significant improvements across all enhanced variants. Each component contributes distinct and complementary benefits: 3D-TPR encoding improves motion consistency (lower FLOLPIPS), texture-focus strategy enhances high-frequency details (higher MUSIQ), and mask uncertainty penalty suppresses occlusion artifacts (lower PI). When combined, the full model yield the best overall performance, confirming that our framework effectively enhances perceptual quality and motion coherence in ZI. Furthermore, when high-quality disparity supervision is available (Appendix A.10), the framework achieves notably larger gains, indicating its strong potential to exploit more accurate geometric priors.

To analyze the effect of optical flow accuracy on 3D-TPR, we downsample the RAFT inputs to 1 $\times$, 2 $\times$, 4 $\times$, and 8 $\times$ resolutions. As RAFT is resolution-sensitive, this setup offers an effective proxy for flow fidelity variation. All estimated flows are bilinearly upsampled to original resolution for training on AMT and IFRNet. As shown in Table 2, lower-resolution flows (4 $\times$ /8 $\times$) lead to noticeable degradation, while 2 $\times$ setting achieves comparable performance to the 1x resolution case while significantly reducing computation. We therefore adopt 2 $\times$ as our default configuration.

4.3 Qualitative results

To demonstrate the visual quality of our synthetic intermediate supervision, Figure 2 shows representative samples from the ZI dataset. I_v achieves smooth transitions in photometrical and geometric structure, while faithfully preserving fine textures and high-frequency details. Despite being generated without ground-truth supervision, the results exhibit strong visual consistency and realism, providing effective guidance for training ZI models.

In Figure 3, we present visual comparisons between 2D and 3D-TPR methods on Vimeo90K. Across multiple FI networks, the 3D-TPR method consistently yields significantly better perceptual quality, producing clearer and sharper images. In contrast, 2D approaches often suffer from noticeable blur, while our 3D-TPR framework maintains fine-grained textures and structural consistency throughout. These results highlight the strong perceptual quality and robustness of the 3D-TPR framework.

Meanwhile, Figure 4 offers a qualitative comparison of ZI results across FI networks and devices. 3D_f-TPR consistently produces higher-quality outputs with fewer artifacts, less blur, and sharper structures. Notably, RIFE, IFRNet, and AMT exhibit significantly reduced blur, while EMA-VFI produces clearer and more coherent line structures. Visual results demonstrate the effectiveness of our data generation method in cross-device generalization for the ZI task. Building upon this, OmniZoom (3D_f-TPR) further exhibits robust performance across diverse FI networks and mobile platforms, offering a universal solution for real-world ZI. Additional visual results are provided in Appendix A.8, and comparisons of 1D and 2D fine-tuning strategies are presented in Appendix A.9.

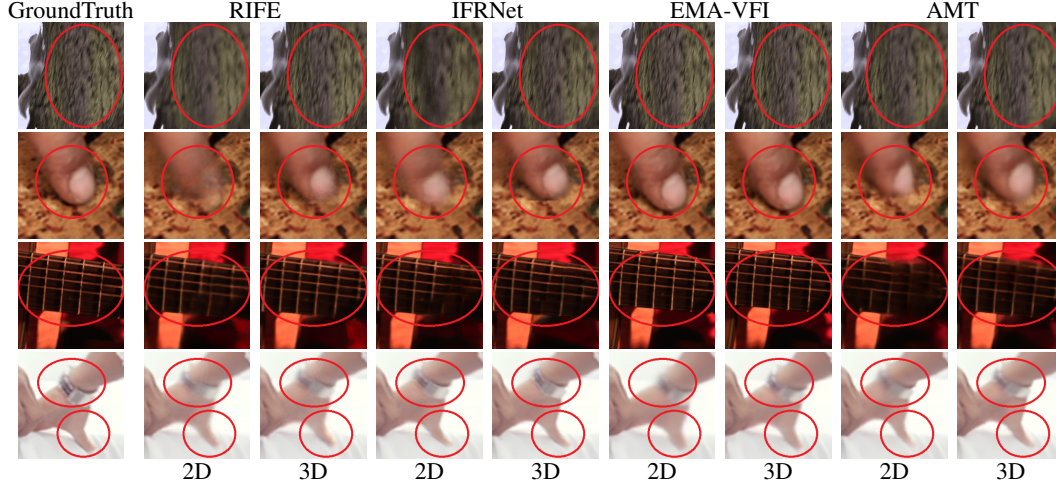


Figure 3: Visual comparison of 2D and 3D-TPR FI results across networks. Each row shows interpolation results at the same timestep.

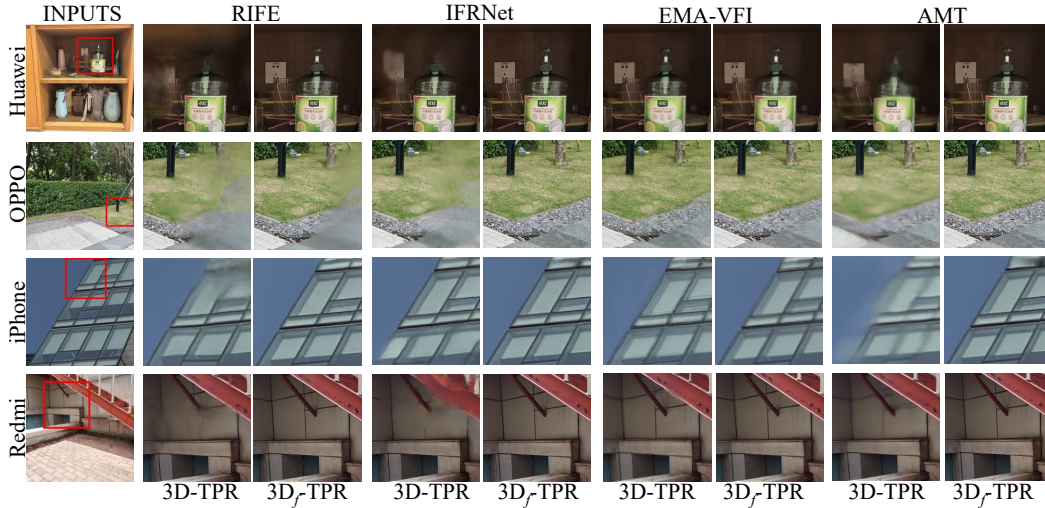


Figure 4: Qualitative results on real-world data across four FI networks at timestep $t = 1/2$. The subscript f denotes models finetuned on our ZI dataset.

4.4 Quantitative results

As shown in Table 3, we conduct quantitative comparisons between the 2D and 3D-TPR frameworks on Vimeo90K, Inter4K, and UCF101. On Vimeo90K, 3D-TPR achieves clear gains in PSNR, SSIM, and LPIPS, indicating better reconstruction and perceptual quality. On the unseen Inter4K and UCF101 benchmarks, 3D-TPR still improves perceptual metrics (SSIM and LPIPS), demonstrating strong generalization. Although pixel-centric metric PSNR may not always improve, ZI focuses on perceptually plausible synthesis rather than pixel-aligned outputs. Also, PSNR is less sensitive to blur [55], which is a common artifact in ZI. Combined with the visual results in Figure 3, it confirms the effectiveness and robustness of 3D-TPR for real-world ZI.

Table 4 offers quantitative results on real-world ZI tasks across four smartphone test sets under four FI networks. Due to space constraints, we use 3D in this table to represent our proposed 3D-TPR framework. 3D-TPR consistently achieves second-best performance across most devices and metrics, clearly outperforming the 2D baseline and demonstrating its effectiveness in handling ZI tasks even without domain-specific tuning. After finetuning on our ZI dataset, OmniZoom (3D_f-TPR) achieves the best results across nearly all models and devices, highlighting both the strong cross-device generalization capability of the dataset and the plug-and-play adaptability of the framework.

Table 3: Comparisons on Vimeo90k, Inter4K and UCF101 using 2D and 3D-TPR FI frameworks. **Bold** indicates the best metric.

Benchmarks	Metrics	RIFE		IFRNet		EMA-VFI		AMT	
		2D	3D-TPR	2D	3D-TPR	2D	3D-TPR	2D	3D-TPR
Vimeo90k	PSNR $\uparrow$	27.40	27.51	27.13	27.21	24.73	24.86	27.17	27.22
	SSIM $\uparrow$	0.901	0.902	0.899	0.901	0.851	0.853	0.902	0.902
	LPIPS $\downarrow$	0.086	0.081	0.078	0.074	0.081	0.080	0.081	0.084
inter4K	PSNR $\uparrow$	33.92	34.06	33.73	33.72	30.06	30.29	33.57	33.80
	SSIM $\uparrow$	0.951	0.952	0.952	0.952	0.903	0.904	0.953	0.955
	LPIPS $\downarrow$	0.048	0.046	0.046	0.045	0.044	0.044	0.048	0.047
UCF101	PSNR $\uparrow$	35.54	35.85	35.42	35.43	36.74	36.65	35.37	35.33
	SSIM $\uparrow$	0.928	0.983	0.984	0.986	0.984	0.985	0.984	0.984
	LPIPS $\downarrow$	0.017	0.018	0.017	0.017	0.012	0.012	0.018	0.017

Table 4: Comparisons on real-world data across four FI models. Subscript f denotes models finetuned on our ZI dataset. **Bold** indicates the best performance, and underline is the second best.

Device	Metrics	RIFE			IFRNet			EMA-VFI			AMT		
		2D	3D	3D $_f$	2D	3D	3D $_f$	2D	3D	3D $_f$	2D	3D	3D $_f$
Huawei	NIQE $\downarrow$	3.8464	<u>3.8651</u>	3.7422	3.6801	3.4852	<u>3.5035</u>	3.6363	<u>3.8470</u>	3.8341	<u>3.5665</u>	3.6459	3.4547
	PI $\downarrow$	4.2505	<u>4.1537</u>	3.9360	3.9570	<u>3.3150</u>	3.2520	3.7240	3.8566	<u>3.8467</u>	<u>3.5147</u>	3.6132	3.3121
	CLIP-IQA $\uparrow$	0.3691	<u>0.4939</u>	0.5233	0.5422	<u>0.5784</u>	0.5909	0.5612	<u>0.5621</u>	0.5684	0.5428	<u>0.5498</u>	0.6018
	MUSIQ $\uparrow$	44.8268	<u>58.8362</u>	60.8585	57.4632	<u>73.0233</u>	74.1220	61.7263	<u>62.7403</u>	63.5048	71.4369	<u>71.7016</u>	73.7092
iPhone	NIQE $\downarrow$	4.2031	<u>4.1027</u>	3.8601	3.8786	3.6923	<u>3.6953</u>	3.8155	<u>4.0555</u>	4.0934	<u>3.5932</u>	3.7053	3.5087
	PI $\downarrow$	4.5340	<u>4.3503</u>	4.0657	4.2090	<u>3.3942</u>	3.3165	3.9994	4.1304	<u>4.1123</u>	<u>3.4293</u>	3.5367	3.2661
	CLIP-IQA $\uparrow$	0.4821	<u>0.5366</u>	0.5492	0.5240	<u>0.5829</u>	0.5949	0.5352	<u>0.5387</u>	0.5503	<u>0.5931</u>	0.5930	0.6211
	MUSIQ $\uparrow$	55.0577	<u>59.6800</u>	60.7988	57.4088	<u>73.3569</u>	73.8292	58.1422	<u>58.6204</u>	60.1002	71.9805	<u>72.1243</u>	73.6150
OPPO	NIQE $\downarrow$	4.6364	<u>4.6100</u>	4.3968	4.5820	5.2039	<u>5.1133</u>	4.5568	<u>4.5166</u>	4.5071	<u>4.9042</u>	5.3017	4.6699
	PI $\downarrow$	5.0848	<u>5.0104</u>	4.6815	4.9749	<u>4.9603</u>	4.7144	4.9569	<u>4.9022</u>	4.8752	<u>4.9676</u>	5.3808	4.5702
	CLIP-IQA $\uparrow$	0.4989	<u>0.5363</u>	0.5830	0.5525	0.5212	<u>0.5461</u>	0.5622	0.5590	<u>0.5591</u>	<u>0.4798</u>	0.4518	0.5595
	MUSIQ $\uparrow$	63.2645	<u>65.7912</u>	67.1494	67.2635	<u>75.2718</u>	75.4545	67.4327	<u>67.5921</u>	67.6444	<u>73.0842</u>	72.1563	75.2187
Redmi	NIQE $\downarrow$	5.0138	<u>4.8764</u>	4.4720	5.0098	<u>4.2837</u>	4.0695	4.4088	<u>4.3424</u>	3.8852	5.1925	<u>4.7426</u>	3.9835
	PI $\downarrow$	5.3615	<u>5.0247</u>	4.5195	5.1108	<u>3.5993</u>	3.3880	4.6708	<u>4.5974</u>	4.5519	5.4335	<u>4.0235</u>	3.3548
	CLIP-IQA $\uparrow$	0.4077	<u>0.4664</u>	0.4930	0.4219	<u>0.4913</u>	0.5152	0.4599	<u>0.4807</u>	0.4947	0.4336	<u>0.4754</u>	0.5383
	MUSIQ $\uparrow$	56.3870	<u>61.9235</u>	63.6990	57.5453	<u>73.7573</u>	74.3871	57.0371	<u>59.6610</u>	60.7579	57.6671	<u>71.9982</u>	74.3325

5 Conclusions

This paper tackles the ZI task that aims to synthesize geometrically and photometrically consistent intermediate frames between dual-camera images. We propose OmniZoom, a universal solution comprising two key components. First, we introduce a cross-device ZI dataset construction pipeline that enables device-agnostic supervision through spatial transition modeling, dynamic color adaptation, and cross-domain consistency learning. Second, we develop a plug-and-play 3D-TPR framework that enhances ZI accuracy and perceptual quality by leveraging 3D trajectory encoding and geometry-aware feature similarity. Experiments on public benchmarks and real-world test sets show that OmniZoom consistently improves performance across diverse FI models and devices. We believe this exploration bridges the gap between pre-trained interpolation models and real-world dual-camera systems, paving the way for more practical and adaptable ZI solutions.

6 Limitations

While OmniZoom is designed to be plug-and-play across various FI networks, achieving optimal performance may still require adjustments to the training schedule, such as tuning the learning rate warm-up strategy, rebalancing loss weights, or modifying batch size. These lightweight adjustments reflect the need for architecture-aware training when applying our framework to diverse settings.

References

- [1] Nisar Ahmed, Hafiz Muhammad Shahzad Asif, and Hassan Khalid. Pqi: perceptual image quality index based on ensemble of gaussian process regression. *Multimedia Tools and Applications*, 80(10):15677–15700, February 2021.
- [2] Hadi Alzayer, Abdullah Abuolaim, Leung Chun Chan, Yang Yang, Ying Chen Lou, Jia-Bin Huang, and Abhishek Kar. Dc2: Dual-camera defocus control by learning to refocus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21488–21497, 2023.
- [3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3703–3712, 2019.
- [4] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):933–948, 2021.
- [5] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.
- [6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.
- [7] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, 2025.
- [8] David J Brady, Michael E Gehm, Ronald A Stack, Daniel L Marks, David S Kittle, Dathon R Golish, EM Vera, and Steven D Feller. Multiscale gigapixel photography. *Nature*, 486(7403):386–389, 2012.
- [9] David J Brady, Wubin Pang, Han Li, Zhan Ma, Yue Tao, and Xun Cao. Parallel cameras. *Optica*, 5(2):127–137, 2018.
- [10] Yang Cheng, Jie Cao, Xin Tang, and Qun Hao. Optical zoom imaging systems using adaptive liquid lenses. *Bioinspiration & Biomimetics*, 16(4):041002, 2021.
- [11] Duolikun Danier, Fan Zhang, and David Bull. Flolpips: A bespoke video quality metric for frame interpolation, 2023.
- [12] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016.
- [13] Shaoyan Gai, Feipeng Da, and Xianqiang Dai. A novel dual-camera calibration method for 3d optical measurement. *Optics and Lasers in Engineering*, 104:126–134, 2018.
- [14] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.
- [15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.
- [16] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: real-time intermediate flow estimation for video frame interpolation. *CoRR*, abs/2011.06294, 2020.
- [17] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pages 624–642. Springer, 2022.
- [18] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.
- [19] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slo-mo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018.
- [20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

- [21] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O’Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)*, 25(3):1058–1066, 2006.
- [22] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale Image Quality Transformer . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5128–5137, Los Alamitos, CA, USA, October 2021. IEEE Computer Society.
- [23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- [24] Diederik Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5. San Diego, California, 2015.
- [25] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2022.
- [26] Haoying Li, Ziran Zhang, Tingting Jiang, Peng Luo, Huajun Feng, and Zhihai Xu. Real-world deep local motion deblurring. In *proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 1314–1322, 2023.
- [27] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023.
- [28] Yi-Hsin Lin, Ming-Syuan Chen, and Hung-Chun Lin. An electrically tunable optical zoom system using two composite liquid crystal lenses with a large zoom ratio. *Optics express*, 19(5):4714–4721, 2011.
- [29] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE international conference on computer vision*, pages 4463–4471, 2017.
- [30] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [32] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.
- [33] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. *CoRR*, abs/2003.05534, 2020.
- [34] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14539–14548, 2021.
- [35] Adam Paszke, Gross, Sam, and Massa. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’ Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [36] Sivalogeswaran Ratnasingam. Deep camera: A fully convolutional neural network for image signal processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.
- [38] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.
- [40] Khurram Soomro, Amir Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 12 2012.

- [41] Alexandros Stergiou and Ronald Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. *CoRR*, abs/2111.00772, 2021.
- [42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [43] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019.
- [44] Joost R. van Amersfoort, Wenzhe Shi, Alejandro Acosta, Francisco Massa, Johannes Totz, Zehan Wang, and Jose Caballero. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *CoRR*, abs/1711.06045, 2017.
- [45] Jianyi Wang, Kelvin Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:2555–2563, 06 2023.
- [46] Tengfei Wang, Jiaxin Xie, Wenxiu Sun, Qiong Yan, and Qifeng Chen. Dual-camera super-resolution with aligned attention modules. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2001–2010, 2021.
- [47] Wen Wang, Qiuyu Wang, Kecheng Zheng, Hao OUYANG, Zhekai Chen, Biao Gong, Hao Chen, Yujun Shen, and Chunhua Shen. Framer: Interactive frame interpolation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [48] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International conference on machine learning*, pages 5180–5189. PMLR, 2018.
- [49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [50] Renlong Wu, Zhilu Zhang, Yu Yang, and Wangmeng Zuo. Dual-camera smooth zoom on mobile phones. In *European Conference on Computer Vision*, pages 250–269. Springer, 2024.
- [51] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023.
- [52] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127:1106–1125, 2019.
- [53] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023.
- [54] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18537–18546, June 2023.
- [55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.
- [57] Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Häne, Sean Fanello, and Rahul Garg. Du 2 net: Learning depth estimation from dual-cameras and dual-pixels. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 582–598. Springer, 2020.
- [58] Zhilu Zhang, Ruohao Wang, Hongzhi Zhang, Yunjin Chen, and Wangmeng Zuo. Self-supervised learning for real-world super-resolution from dual zoomed observations. In *European Conference on Computer Vision*, pages 610–627. Springer, 2022.
- [59] Yang Zhao, Ronggang Wang, Wei Jia, Wangmeng Zuo, Xiaoping Liu, and Wen Gao. Deep Reconstruction of Least Significant Bits for Bit-Depth Expansion. *IEEE Transactions on Image Processing*, 28(6):2847–2859, June 2019.

- [60] Zhihang Zhong, Gurunandan Krishnan, Xiao Sun, Yu Qiao, Sizhuo Ma, and Jian Wang. Clearer frames, anytime: Resolving velocity ambiguity in video frame interpolation. In *European Conference on Computer Vision*, pages 346–363. Springer, 2024.
- [61] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 286–301. Springer, 2016.

A Appendix

The content of the supplementary material involves:

- Details of ColorNet in Appendix A.1.
- Dual-camera data sources for the ZI dataset in Appendix A.2.
- Details of the synthetic ZI dataset and computational cost in Appendix A.3.
- Formula reasoning of 3D-TPR framework in Appendix A.4.
- Comparative evaluation of virtual view synthesis method for ZI task in Appendix A.5.
- Additional experiment on disparity map sources in Appendix A.6.
- Details of the real-world multi-device ZI test sets in Appendix A.7.
- Additional visual results of OmniZoom on the real-world ZI test sets in Appendix A.8.
- Benchmarking ZI dataset across 1D, 2D, and 3D-TPR frameworks in Appendix A.9.
- Upper bound analysis of 3D-TPR framework in Appendix A.10.
- Additional visual results of 3D-TPR framework in Appendix A.11.
- Broader impacts in Appendix A.12.

We provide a project page at <https://omnizoom.github.io/OmniZoom/>, where visual results are available, and the code/dataset will be released soon.

A.1 Details of ColorNet

A.1.1 Architecture of ColorNet

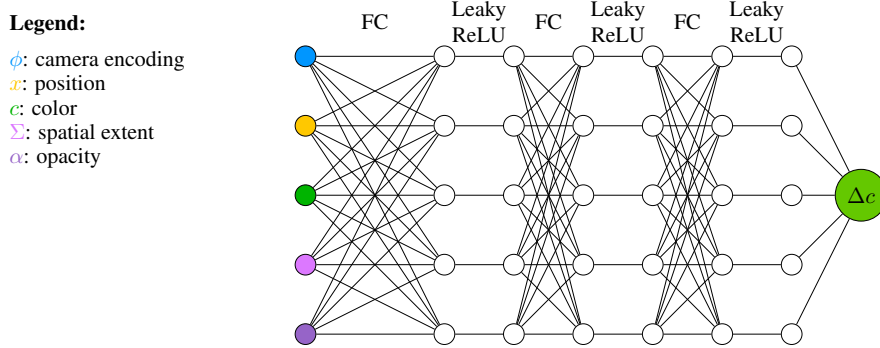


Figure 5: Architecture of ColorNet.

As illustrated in Figure 5, ColorNet is a lightweight multilayer perceptron (MLP) designed to estimate color residuals Δc for neural 3D rendering. The network takes as input a five-dimensional feature tuple: the camera encoding ϕ , the center position x , the base color c , the anisotropic covariance Σ , and the opacity α . These inputs are first projected into a latent feature space through a fully connected (FC) layer, which is then processed by three consecutive FC blocks, each followed by a LeakyReLU activation to ensure non-linearity and mitigate the risk of neuron inactivation. Each FC layer contains 5 hidden units. Finally, the output layer maps the intermediate representation to a 3-dimensional color offset vector Δc via a linear transformation. This residual is then added to the original color c , yielding the refined output color used for rendering.

A.1.2 Ablations of ColorNet

We perform a no-reference image quality comparison between our proposed ColorNet and several common photometric calibration baselines: (1) *No Calibration*, (2) *Mean-Std Normalization* [48], and (3) *Reinhard Color Transfer* [51]. The evaluations are conducted on rendered intermediate frames using four widely adopted metrics: NIQE, PI, CLIP-IQA, and MUSIQ.

Table 5: No-reference image quality comparison of ColorNet against common photometric calibration baselines on rendered intermediate frames. Lower NIQE and PI indicate better perceptual quality, while higher CLIP-IQA and MUSIQ denote better alignment with human perception. ColorNet consistently achieves the best performance across all metrics, demonstrating its effectiveness in modeling fine-grained photometric variations.

Method	NIQE ↓	PI ↓	CLIP-IQA ↑	MUSIQ ↑
No Calibration	3.5832	3.8378	0.5193	68.4036
Mean-Std Norm	3.3193	3.7757	0.4577	63.0220
Reinhard Transfer	3.2357	3.7345	0.4588	64.3218
ColorNet (Ours)	2.8885	3.1515	0.6099	69.3328



Figure 6: Representative dual-camera data sources. The top row displays samples from HuaweiPura70Ultra, while the bottom row presents examples from the publicly available RedmiK50Ultra dataset [50]. Each scene contains paired wide-angle (I_w) and main-camera (I_m) images, covering diverse content and geometric configurations.

As reported in Table 5, **ColorNet consistently outperforms all baseline methods**, demonstrating its superior capability in capturing fine-grained photometric variations that simpler approaches fail to model. These results highlight the effectiveness of ColorNet in enhancing color consistency and perceptual quality across intermediate frames.

A.2 Dual-camera data sources for the ZI dataset

As mentioned in the main paper, our ZI dataset comprises dual-camera zoom image pairs from HuaweiPura70Ultra and RedmiK50Ultra. Specifically, we collect Huawei data ourselves across a variety of indoor and outdoor scenarios, while the Redmi sequences are adopted from the publicly available dataset introduced in [50]. The image resolutions vary by device: 2133×1600 for Huawei and 1632×1224 for Redmi, consistent with their native imaging pipelines.

In total, the combined dataset covers 132 real-world scenes, consisting of 79 indoor and 53 outdoor environments. Indoor scenes include structured environments such as classrooms, shopping malls, and cafeterias, while outdoor scenes cover playgrounds, parks, and other open spaces with greater geometric variation. For each scene, we capture 15 pairs of main–wide camera images, along with 6 additional wide-angle views from peripheral viewpoints to enhance spatial diversity. These supplementary images increase the geometric baseline and play a key role in stabilizing 3DGS optimization. Representative examples from both devices are shown in Figure 6.

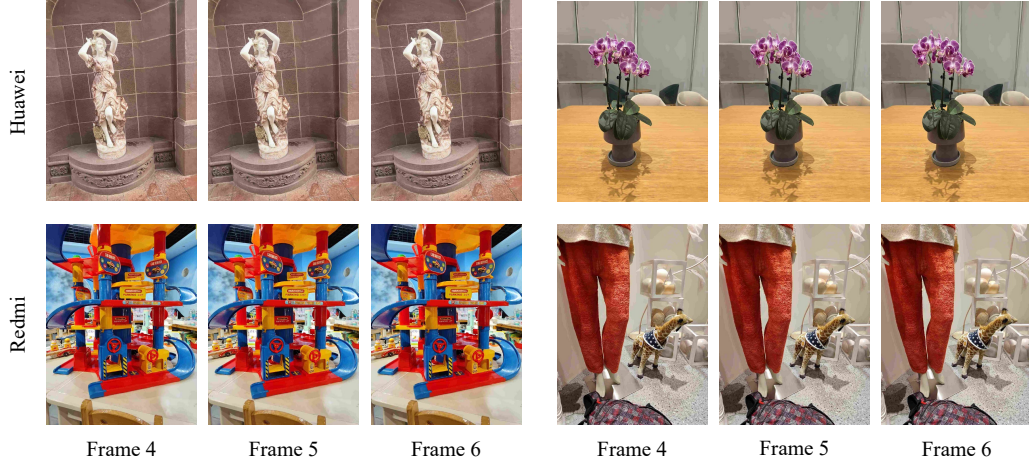


Figure 7: Representative frames (Frames 4, 5, and 6) from our synthetic ZI dataset. Each row illustrates three temporally adjacent frames sampled from a 16-frame virtual zoom sequence, demonstrating smooth transitions in both appearance and geometry. Examples are shown from Huawei and Redmi devices, covering diverse indoor and outdoor scenes with varying structural complexity.

A.3 Details of the synthetic ZI dataset and computational cost

We generate 205 synthetic zoom sequences, each consisting of 16 interpolated frames. These sequences serve as high-quality supervision for training and evaluation in ZI tasks. Representative examples of the rendered dataset are shown in Figure 7.

The training duration for each scene depends on its visual complexity and structural content, with an average of approximately 1.5 hours per scene. Rendering a complete zoom sequence with 16 interpolated frames typically takes less than 20 seconds. All procedures are conducted on a workstation with 40 GB of memory, consistent with our main experimental setup.

A.4 Formula reasoning of 3D-TPR framework

A.4.1 Mathematical modeling and depth constraints for projective ambiguity

Modeling the projection from 3D space to 2D image plane: Let $\mathbf{P} = (X, Y, Z)^T$ denote a 3D point in the scene, which is projected onto 2D image coordinates (x, y) via the pinhole camera model. The projection is formulated as:

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}; \quad \pi(\mathbf{P}) = \left(\frac{fX}{Z}, \frac{fY}{Z} \right)^T, \quad (19)$$

where π denotes the 3D-to-2D projection function, and f represents the camera’s focal length. When an object moves through 3D space, its instantaneous displacement over a time interval Δt is governed by its 3D velocity vector $\mathbf{v} = (v_X, v_Y, v_Z)^T$:

$$\Delta \mathbf{P} = \mathbf{v} \Delta t = (v_X \Delta t, v_Y \Delta t, v_Z \Delta t)^T, \quad (20)$$

where $\Delta \mathbf{P}$ denotes the change in 3D position during Δt . Let $I(\mathbf{P}, t)$ denote the projection of a 3D point $\mathbf{P} = (X, Y, Z)^T$ onto the image plane at time t . When the object is in motion, the position of $\mathbf{P}$ evolves over time as:

$$\mathbf{P}(t + \Delta t) = \mathbf{P}(t) + \Delta \mathbf{P} = \mathbf{P}(t) + \mathbf{v} \Delta t. \quad (21)$$

The corresponding image plane displacement $\delta = (\Delta x, \Delta y)$ induced by 3D motion can be approximated via a first-order Taylor expansion:

$$\delta = \pi(\mathbf{P} + \mathbf{v} \Delta t) - \pi(\mathbf{P}) \approx \frac{\partial \pi}{\partial \mathbf{P}} \mathbf{v} \Delta t, \quad (22)$$

where $\frac{\partial \pi}{\partial \mathbf{P}}$ is the projection Jacobian matrix J_π , which characterizes how variations in 3D positions affect the projected 2D coordinates. The term $\mathbf{v}\Delta t$ represents the instantaneous 3D displacement, and its transformation through J_π yields the corresponding image plane motion. By substituting the partial derivatives of the projection function, the Jacobian matrix J_π can be explicitly written as:

$$J_\pi = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{fX}{Z^2} \\ 0 & \frac{f}{Z} & -\frac{fY}{Z^2} \end{bmatrix}. \quad (23)$$

Equation 23 constitutes an underdetermined system of equations, whose solution space forms a straight line. Mathematically, this can be expressed as:

$$\mathbf{v} = \mathbf{v}_0 + k\mathbf{n}, \quad \mathbf{n} = \left(\frac{X}{Z}, \frac{Y}{Z}, 1 \right)^T, \quad (24)$$

where $\mathbf{v}_0$ represents a particular solution, $\mathbf{n}$ denotes the direction corresponding to the viewing ray, and $k \in \mathbb{R}$ denotes an arbitrary real number. This formulation implies that infinitely many 3D velocity vectors, which differ only in their components along the line of sight, can result in the same 2D image plane displacement. As a concrete example, consider a 3D scene point located at $\mathbf{P} = (f, 0, Z)^T$:

Scene 1. Suppose the object moves purely along the horizontal axis with the velocity vector $\mathbf{v}_1 = (1, 0, 0)^T$. The resulting image plane displacement δ_1 can be approximated via the projection Jacobian J_π as:

$$\delta_1 = J_\pi \cdot \mathbf{v}_1 \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \cdot 1 + 0 \cdot 0 + \left(-\frac{fX}{Z^2} \right) \cdot 0 \\ 0 \cdot 1 + \frac{f}{Z} \cdot 0 + \left(-\frac{fY}{Z^2} \right) \cdot 0 \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \\ 0 \end{bmatrix} \cdot \Delta t. \quad (25)$$

Scene 2. Now consider the case where the object moves along the optical axis (i.e., in depth), with the velocity vector $\mathbf{v}_2 = (0, 0, -Z/f)^T$. For the 3D point $\mathbf{P} = (f, 0, Z)^T$, the resulting image plane displacement δ_2 is computed as:

$$\delta_2 = J_\pi \cdot \mathbf{v}_2 \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \cdot 0 + 0 \cdot 0 + \left(-\frac{fX}{Z^2} \right) \cdot \left(-\frac{Z}{f} \right) \\ 0 \cdot 0 + \frac{f}{Z} \cdot 0 + \left(-\frac{fY}{Z^2} \right) \cdot \left(-\frac{Z}{f} \right) \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{bmatrix} \cdot \Delta t = \begin{bmatrix} \frac{f}{Z} \\ 0 \end{bmatrix} \cdot \Delta t. \quad (26)$$

The distinct 3D motions illustrated in **Scene 1** and **Scene 2** become indistinguishable when projected onto the 2D image plane, thereby *leading to the non-uniqueness of motion estimation and highlighting the necessity of depth-direction constraints*. By enforcing a constraint along the depth direction, the solution space is reduced from a continuous line to a single, physically plausible point, effectively eliminating the ambiguity and suppressing implausible variations along the depth axis.

A.4.2 Gradient reallocation with texture-focus strategy

Let $\mathcal{S}_T = (x, y) \mid \mathbf{M}(x, y) = 1$ denote the set of texture pixels, where $\mathbf{M}$ is the binary mask defined in Equation 14, and let $\mathcal{S}_C$ denote its complement. According to Equation 16, the expected value of gradient norm over texture regions satisfies the following for $\lambda > 0$:

$$\mathbb{E} \left[\|\nabla_{\theta} \mathcal{L}_{\text{texture-focus}}\|_{\mathcal{S}_T} \right] = (1 + \lambda) \cdot \mathbb{E} \left[\|\nabla_{\theta} \mathcal{L}_{\text{base}}\|_{\mathcal{S}_T} \right]. \quad (27)$$

This formulation indicates that the texture-focus loss reweights the gradient flow to amplify supervision in structurally informative regions. By scaling the gradient magnitude in $\mathcal{S}_T$ by a factor of $(1 + \lambda)$, the network is encouraged to prioritize learning from texture-rich areas, which are typically more perceptually salient and semantically meaningful.

While expected value of the gradient norm within the texture region $\mathcal{S}_T$ is magnified by a factor of $(1 + \lambda)$, the gradient norm in the non-texture region $\mathcal{S}_C$ remains unchanged:

$$\mathbb{E} \left[\|\nabla_{\theta} \mathcal{L}_{\text{texture-focus}}\|_{\mathcal{S}_C} \right] = \mathbb{E} \left[\|\nabla_{\theta} \mathcal{L}_{\text{base}}\|_{\mathcal{S}_C} \right]. \quad (28)$$

Consequently, the texture-focus strategy effectively redistributes the gradient energy in the parameter space by selectively amplifying directions associated with texture regions, thereby enhancing supervision in structurally informative areas. Under momentum-based stochastic gradient descent (SGD), the network parameters are updated as:

$$\theta_{k+1} = \theta_k - \eta \mathbf{v}_k, \quad \mathbf{v}_k = \beta \mathbf{v}_{k-1} + (1 - \beta) \nabla_{\theta} \mathcal{L}_{\text{texture-focus}}(\theta_k), \quad (29)$$

where θ_k denotes the network parameters at the k -th iteration, and θ_{k+1} represents the updated parameters after applying a weighted update in the direction of the momentum term $\mathbf{v}_k$. Here, η is the learning rate, and $\mathbf{v}_k$ is the accumulated momentum, which integrates the current gradient with past updates. We set the momentum coefficient to $\beta = 0.9$, which determines the proportion of the previous momentum $\mathbf{v}_{k-1}$ retained in the current step. The gradient term $\nabla_{\theta} \mathcal{L}_{\text{texture-focus}}(\theta_k)$ is computed with respect to the texture-focus loss at iteration k .

Since the gradient of $\mathcal{L}_{\text{texture-focus}}$ within the texture region $\mathcal{S}_T$ is scaled by a factor of $(1 + \lambda)$, the corresponding entries in the gradient tensor exhibit proportionally larger magnitudes. This induces an optimization bias toward texture-dense regions, thereby encouraging edge sharpening and the recovery of high-frequency details. In contrast, gradients within the non-texture region $\mathcal{S}_C$ remain unchanged. This targeted reallocation strategy prioritizes learning in structurally salient areas, ultimately enhancing the perceptual fidelity and fine-detail quality of the interpolated frames.

Our proposition formally demonstrates that explicitly amplifying momentum components along high-frequency texture directions offers two key benefits: (a) it reduces the number of iterations required to reach local optima by modulating the effective learning rate, and (b) it suppresses oscillatory behaviors in the loss surface caused by the competing objectives of texture underfitting and background overfitting.

Under the flat-minima hypothesis for over-parameterized networks, this gradient reallocation strategy further flattens the loss landscape during finetuning, acting as an implicit regularizer that improves generalization to unseen texture distributions through curvature-driven optimization dynamics. Unlike perceptual losses that entangle high- and low-frequency signals within the shared feature space of pretrained encoders, our texture-focus strategy achieves frequency disentanglement via pixel-level gradient modulation in an encoder-agnostic manner, allowing more precise control over structural detail reconstruction only with an ancillary supervision.

A.4.3 Additional proof of mask penalty strategy

Normalization coefficient $4 \cdot \mathbf{M}(1 - \mathbf{M}) \in [0, 1]$, reaching the maximum peak value at $\mathbf{M} = 0.5$. As $\mathbf{M} \rightarrow 0$ or $\rightarrow 1$, the normalization coefficient value tends to zero. From a Bayesian perspective, the mask value $\mathbf{M}_p^{(i)}$ is modelled as the parameter $\theta_p^{(i)}$ of the posterior Bernoulli distribution $\mathcal{Q}$:

$$(\mathbf{M}_p^{(i)} = \theta_p^{(i)}) \sim (\mathcal{Q} = \text{Bernoulli}(\theta_p^{(i)})), \quad (30)$$

where $\mathbf{M}_p^{(i)}$ represents the model-inferred confidence at pixel position p of the i -th scale, which is calculated by the middle layer of the network and equivalent to $\theta_p^{(i)}$. $\theta_p^{(i)}$ is the parameter of the Bernoulli distribution, representing the probability that the mask value is 1 at position p , indicating the network’s confidence that the current pixel is dominated by the features from I_0 .

The case $\theta_p^{(i)} = 1$ indicates that the mask value at position p is fully dominated by I_0 , whereas $\theta_p^{(i)} = 0$ implies complete dominance by I_1 . $\theta_p^{(i)} = 0.5$ shares the equal contributions from both two reference frames.

The interpolation network dynamically adjusts the weights by learning the distribution of $\theta_p^{(i)}$. Since $\theta_p^{(i)}$ is directly adopted as the parameter of the Bernoulli distribution, its entropy $\mathcal{H}(\theta_p^{(i)})$ quantifies the uncertainty in the fusion decision:

$$\mathcal{H}(\mathcal{Q}) = \mathcal{H}(\theta_p^{(i)}) = -\theta_p^{(i)} \log \theta_p^{(i)} - (1 - \theta_p^{(i)}) \log(1 - \theta_p^{(i)}). \quad (31)$$

A higher entropy value indicates greater uncertainty in the network’s fusion decision at the current pixel, typically when $\theta_p^{(i)}$ approaches 0.5. In contrast, lower entropy represents higher confidence, corresponding to $\theta_p^{(i)}$ values close to 0 or 1. Assume that an implicit Beta prior distribution $\mathcal{P}$ is applied to $\theta_p^{(i)}$:

$$\mathcal{P}(\theta_p^{(i)}) \propto \theta_p^{\alpha-1} (1 - \theta_p)^{\beta-1}, \quad (32)$$

where we adopt a symmetric, U-shaped configuration by setting $\alpha = \beta < 1$. The probability density is higher at $\theta_p = 0$ or $\theta_p = 1$ and lower at median $\theta_p = 0.5$, thereby encoding prior knowledge that induces a binarization tendency. Minimizing the KL divergence between the posterior distribution $\mathcal{Q}$ and the prior distribution $\mathcal{P}$ yields:

$$D_{KL}(\mathcal{Q}||\mathcal{P}) = \mathcal{H}(\mathcal{Q}, \mathcal{P}) - \mathcal{H}(\mathcal{Q}), \quad (33)$$

where $\mathcal{H}(\mathcal{Q}, \mathcal{P}) = -\mathbb{E}_{\mathcal{Q}}[\log \mathcal{P}(\theta_p^{(i)})]$ measures the cross-entropy between the posterior distribution $\mathcal{Q}$ and the prior distribution $\mathcal{P}$, reflecting how well the posterior aligns with the prior. A smaller value indicates that $\mathcal{Q}$ aligns more closely with the distributional preferences encoded in $\mathcal{P}$. $\mathcal{H}(\mathcal{Q})$ quantifies the intrinsic uncertainty of $\mathcal{Q}$, with a higher value indicating increased ambiguity in decision-making.

By designing $\mathcal{L}_{mask}^{(i)} \propto \theta_p^{(i)}(1 - \theta_p^{(i)})$, as derived in Equation 18, we implicitly impose a gradient direction consistent with the low-entropy Beta prior distribution $\mathcal{P}$. When $\theta_p^{(i)}$ approaches to 0.5, the gradient of $\mathcal{H}(\mathcal{Q}, \mathcal{P})$ dominates the optimization direction since $\log \mathcal{P}(\theta_p^{(i)})$ tends to be negative infinity. In that case, $\theta_p^{(i)}$ will be forced to flee quickly from high-uncertainty areas, in alignment with the gradient of $\mathcal{L}_{mask}^{(i)}$. Once $\theta_p^{(i)}$ enters the vicinity of 0 or 1, the gradient of $\mathcal{H}(\mathcal{Q}, \mathcal{P})$ decays, and the gradient of $\mathcal{H}(\mathcal{Q})$ begins to dominate. At this point, the gradient of $\mathcal{L}_{mask}^{(i)}$ remains persistently operative throughout the optimization process, thus ensuring stable convergence of the model. Consequently, minimizing $\mathcal{L}_{mask}^{(i)}$ effectively approximates the minimizing KL divergence $D_{KL}(\mathcal{Q}||\mathcal{P})$. This design is theoretically anchored in the Bayesian framework and promotes training stability through a principled simplification of the target loss.

To sum up, the overall training target loss is:

$$\mathcal{L}_{overall} = \alpha_T \cdot \mathcal{L}_{texture-focus} + \alpha_m \cdot \sum_{i=1}^S \mathcal{L}_{mask}^{(i)} + \alpha_o \cdot \mathcal{L}_{original}, \quad (34)$$

where α_T , α_m , and α_o denote the weighting coefficients for the texture-focus loss, mask regularization loss, and baseline supervision loss, respectively. These weights can be manually specified or adaptively optimized during training, depending on the architectural design and optimization dynamics of the model.

A.5 Comparative evaluation of virtual view synthesis method for ZI task

A.5.1 Comparison of virtual frame generation quality

To evaluate the effectiveness of synthetic supervision for the ZI task, we compare three virtual frame generation methods: ZoomGS [50], 3DGS [23], and our proposed pipeline. For a fair comparison, all baselines are reproduced using their official implementations and default hyperparameters. As shown in Figure 8, our method yields noticeably higher-quality intermediate views, exhibiting sharper textures, reduced artifacts, and more accurate geometric consistency. In contrast, ZoomGS and 3DGS often exhibit geometric distortion and texture degradation, particularly in scenes with complex spatial layouts or high-frequency surface details. These findings highlight the superiority of our pipeline in producing visually reliable and semantically aligned supervision tailored for ZI training.

A.5.2 Evaluating virtual supervision via ZI finetuning

To assess the effectiveness of our synthetic supervision in real-world ZI scenarios, we conduct a downstream finetuning evaluation against two baselines: ZoomGS [50] and 3DGS [23]. For each method, the corresponding virtual dataset is used to finetune four representative FI networks, all integrated into our unified 3D-TPR framework. The resulting models are then evaluated on real-world test sets captured from four distinct smartphone platforms, enabling a comprehensive assessment of cross-device generalization performance.

Quantitative results. Table 6 reports results on Huawei and Redmi devices, both of which are included in all three synthetic ZI datasets. Our method consistently outperforms ZoomGS and 3DGS across all networks and metrics, demonstrating strong supervision quality in seen-device scenarios. While ZoomGS performs reasonably due to its device-specific training coverage, it lacks explicit spatial and color modeling. In contrast, our spatial transition and dynamic color adaptation modules effectively handle cross-camera misalignment and photometric variation, resulting in sharper textures and more coherent frame synthesis, reflected by gains in CLIP-IQA and MUSIQ. 3DGS also employs Huawei and Redmi data, but builds on monocular reconstruction assumptions and ignores dual-camera geometric or ISP discrepancies. Consequently, the composite supervision shows less consistency, which universally degrades finetuning efficacy on all tasks.



Figure 8: Visual comparison of virtual frame generation methods for ZI supervision. Compared to ZoomGS [50] and 3DGS [23], our method produces intermediate frames with fewer artifacts and better detail preservation, demonstrating superior supervision quality for training ZI models.

Table 6: Quantitative comparison on real-world dual-camera test sets captured by Huawei and Redmi devices, which are included in the synthetic dataset construction. Each FI model is evaluated under four supervision variants: original 3D-TPR pretrained (Base), and finetuned on 3DGS, ZoomGS, and our proposed supervision f . **Bold** indicates the best result, and underline denotes the second best.

Networks	Methods	HuaweiPura70Pro				RedmiK50Ultra			
		NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑	NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑
RIFE	Base	3.8651	4.1537	0.4939	58.8362	4.8764	5.0247	0.4664	61.9235
	3DGS _f	3.8309	4.2570	<u>0.5183</u>	<u>60.2724</u>	4.8457	5.1787	0.3459	52.4562
	ZoomGS _f	<u>3.7902</u>	<u>4.0084</u>	0.5013	59.0780	<u>4.5598</u>	<u>4.7340</u>	<u>0.4716</u>	61.2694
	Ours _f	3.7422	3.9360	0.5233	60.8585	4.4720	4.5195	0.4930	63.6990
IFRNet	Base	3.4852	3.3150	0.5784	73.0233	4.2837	3.5993	0.4913	73.7573
	3DGS _f	3.5310	<u>3.2570</u>	0.5183	70.2725	4.2232	3.5776	0.4259	70.0659
	ZoomGS _f	3.5132	3.2769	<u>0.5867</u>	<u>74.0218</u>	<u>4.0879</u>	<u>3.3928</u>	<u>0.5127</u>	<u>74.2312</u>
	Ours _f	<u>3.5035</u>	3.2520	0.5909	74.1220	4.0695	3.3880	0.5152	74.3871
EMA-VFI	Base	3.8470	3.8566	<u>0.5621</u>	<u>62.7403</u>	4.3424	4.5974	0.4807	59.6610
	3DGS _f	3.8888	3.4184	0.5540	62.7141	4.2464	4.4593	<u>0.4908</u>	59.2312
	ZoomGS _f	3.6360	3.8499	0.5229	61.2508	<u>4.2106</u>	<u>4.4610</u>	0.4718	<u>60.1914</u>
	Ours _f	<u>3.8341</u>	<u>3.8467</u>	0.5684	63.5048	3.8852	4.5519	0.4947	60.7579
AMT	Base	3.6459	3.6132	0.5498	71.7016	4.7426	4.0235	0.4754	71.9982
	3DGS _f	3.4894	3.3929	0.5184	70.5856	4.3934	3.7834	0.4646	71.4254
	ZoomGS _f	<u>3.4593</u>	<u>3.3395</u>	<u>0.5847</u>	<u>73.7058</u>	<u>4.0653</u>	<u>3.4472</u>	<u>0.5209</u>	74.4339
	Ours _f	3.4547	3.3121	0.6018	73.7092	3.9835	3.3548	0.5383	<u>74.3325</u>

Table 7 evaluates generalization to iPhone and OPPO: two unseen devices that are not involved in training or synthetic data generation. Despite the domain gap, our method consistently achieves the best performance across all networks and evaluation metrics. In contrast, ZoomGS and 3DGS exhibit clear performance degradation due to their limited capacity to model cross-device geometric and

photometric variations. These results highlight the strong generalization ability of our pipeline, which provides robust, device-agnostic supervision across heterogeneous smartphone platforms.

Our method consistently outperforms existing virtual frame generation pipelines across both seen and unseen device scenarios. These gains are attributed to the explicit modeling of cross-device geometric

Table 7: Quantitative evaluation of cross-device generalization on iPhone and OPPO, two unseen devices excluded from the synthetic dataset generation. Despite the domain shift, models finetuned with our supervision consistently outperform those trained with 3DGS and ZoomGS. **Bold** indicates the best result, and underline denotes the second best.

Networks	Methods	iPhone16ProMax				OPPOFindX8Ultra			
		NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑	NIQE↓	PI↓	CLIP-IQA↑	MUSIQ↑
RIFE	Base	4.1027	4.3503	0.5366	<u>59.6800</u>	4.6100	5.0104	0.5366	65.7912
	3DGS _f	3.9618	4.2914	0.4687	54.5013	4.4927	4.9957	0.4408	60.6637
	ZoomGS _f	<u>3.8950</u>	<u>4.1679</u>	<u>0.5411</u>	59.6339	<u>4.4543</u>	<u>4.8111</u>	<u>0.5779</u>	<u>66.6404</u>
	Ours _f	3.8601	4.0657	0.5492	60.7988	4.3968	4.6815	0.5830	67.1494
IFRNet	Base	<u>3.6923</u>	3.3942	0.5829	73.3569	5.2039	4.9603	0.5212	75.2718
	3DGS _f	3.6233	<u>3.3400</u>	0.5583	71.7621	<u>5.1306</u>	4.8809	0.4755	74.3894
	ZoomGS _f	3.7501	3.3498	<u>0.5941</u>	<u>73.7598</u>	5.1630	<u>4.7927</u>	<u>0.5364</u>	<u>75.3461</u>
	Ours _f	3.6953	3.3165	0.5949	73.8292	5.1133	4.7144	0.5461	75.4545
EMA-VFI	Base	4.0555	4.1304	0.5387	58.6204	<u>4.5166</u>	4.9022	0.5590	67.5921
	3DGS _f	3.6258	3.8183	<u>0.5418</u>	59.2020	4.8225	4.9720	0.5868	66.8282
	ZoomGS _f	<u>3.8068</u>	<u>3.9542</u>	0.5377	<u>59.5872</u>	4.5455	<u>4.8818</u>	0.5551	68.2806
	Ours _f	4.0934	4.1123	0.5503	60.1002	4.5071	4.8752	<u>0.5591</u>	<u>67.6444</u>
AMT	Base	3.7053	3.5367	0.5930	<u>72.1243</u>	5.3017	5.3808	0.4518	72.1563
	3DGS _f	3.6991	3.4205	0.5644	71.6435	4.9154	5.0089	0.4639	72.5701
	ZoomGS _f	<u>3.5192</u>	<u>3.4044</u>	<u>0.5985</u>	71.8655	<u>4.6889</u>	<u>4.7544</u>	<u>0.5581</u>	<u>72.6175</u>
	Ours _f	3.5087	3.2661	0.6211	73.6150	4.6699	4.5702	0.5595	75.2187

transitions and photometric inconsistencies, two key factors for ensuring structurally accurate and perceptually consistent zoom interpolation in real-world.

Qualitative results. We further perform a qualitative comparison of interpolation outcomes across four FI networks and four smartphone platforms in real-world ZI tasks. As shown in Figure 9, we visualize the results of models finetuned with three types of synthetic supervision: ZoomGS, 3DGS, and our proposed approach, alongside the original 3D-TPR pretrained baselines without finetuning. ZoomGS_f performs reasonably on Huawei and Redmi devices, which are included in its training dataset. However, it still suffers from noticeable blurring and diminished structural fidelity, particularly around object boundaries and fine textures. 3DGS_f, which lacks explicit device-specific modeling, exhibits limited generalization and introduces geometric distortions and color artifacts across all devices. In contrast, our method consistently generates sharper structures, more coherent textures, and significantly fewer artifacts. For instance, text contours appear more defined on Huawei, linear boundaries are better preserved on Redmi, and perceptual blurring is substantially reduced on OPPO and iPhone. These qualitative observations further underscore the importance of modeling device geometric and photometric characteristics, and highlight the superior visual fidelity enabled by our synthetic supervision pipeline.

Overall. Both quantitative and qualitative results consistently demonstrate the superiority of our synthetic supervision across a wide range of FI networks and smartphone platforms. By explicitly modeling cross-device geometric misalignment and photometric variation, our method enables more effective finetuning, yielding improvements in perceptual quality, structural fidelity, and generalization to unseen devices.

A.6 Additional ablation experiment on disparity map sources

To rigorously evaluate the robustness of disparity cues in our proposed 3D-TPR encoding, we systematically ablate the disparity map sources that constitute the critical foundation for similarity computation. Employing EMA-VFI[53] backbone with RAFT-based warping[42], we conduct comparative analyses between our lightweight Lite-Mono[54] architecture and the sophisticated DepthPro[7]. The experimental results are shown in Table 8.

It is crucial to emphasize that this ablation study does not directly assess the disparity variation vector itself, but rather evaluates the quality and reliability of source similarity maps utilized in constructing the 3D trajectory vectors. Experimental results on the Vimeo90K benchmark demonstrate that both

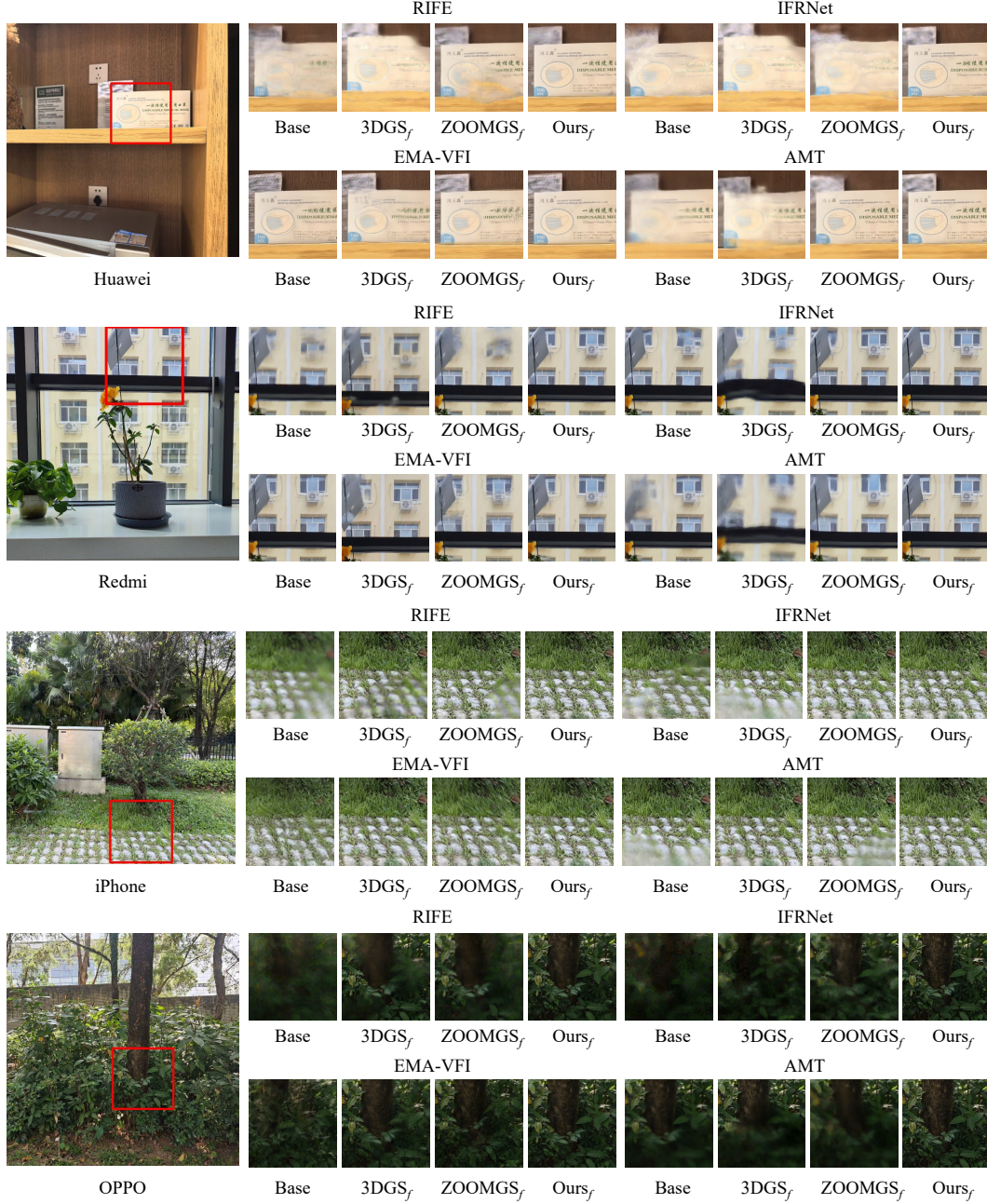


Figure 9: Qualitative comparison of finetuned interpolation results on real-world ZI tasks across four smartphone platforms and four FI networks. We compare models finetuned with virtual supervision from 3DGS, ZoomGS, and our proposed method, alongside unfinetuned baselines ("Base") implemented within the unified 3D-TPR framework. Our method consistently produces sharper structures and fewer artifacts. Notably, text contours are better preserved on Huawei, boundary lines are more clearly delineated on Redmi, and perceptual blurring is substantially reduced on iPhone and OPPO.

Table 8: Ablation study on disparity map sources.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
2D	24.73	0.851	0.081
3D-TPR(Lite-Mono)	24.86	0.853	0.080
3D-TPR(Depth-pro)	24.90	0.852	0.079

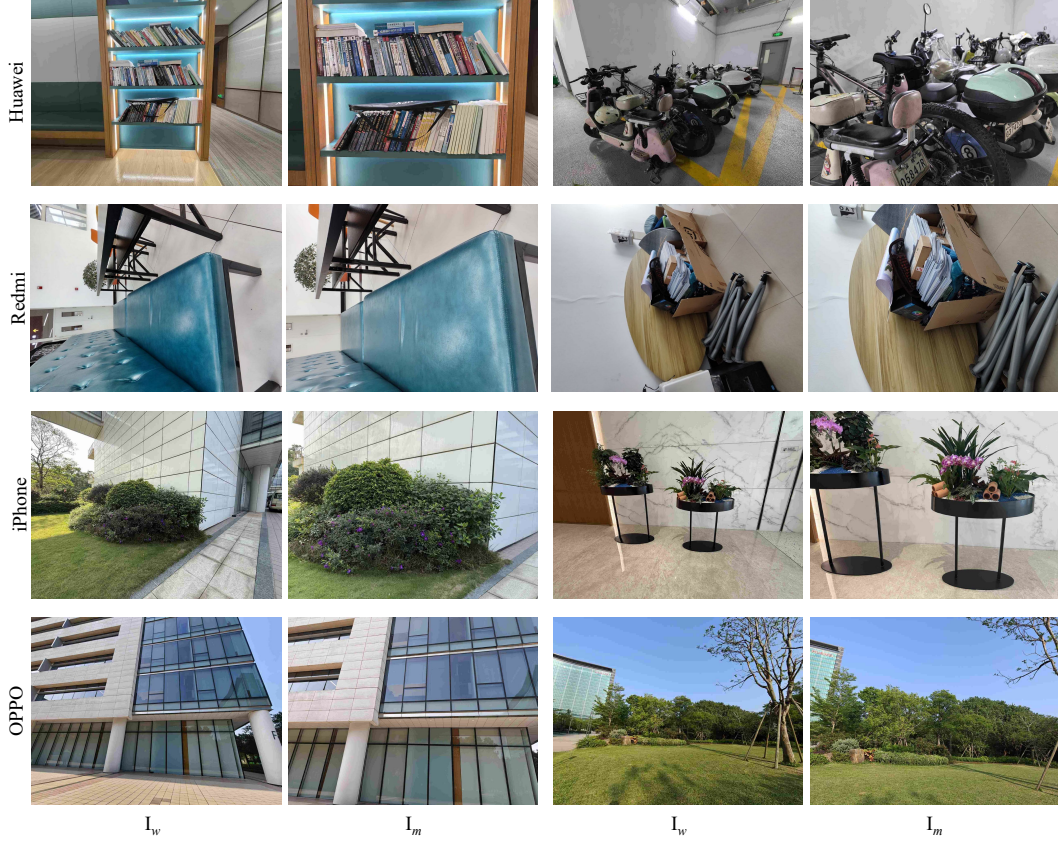


Figure 10: Representative dual-camera samples from our real-world multi-device ZI test sets. Each row corresponds to a different smartphone platform (Huawei, Redmi, iPhone and OPPO), showing paired zoom images captured under diverse scenes and lighting conditions. For layout consistency, RedmiK50Ultra samples are rotated 90 degrees counterclockwise.

3D-TPR implementations consistently surpass the 2D baseline, validating the stability and generalizability of our motion priors across different disparity estimators. Notably, while DepthPro[7] generates perceptually sharper outputs, its computational complexity yields only marginal performance gains. Consequently, we adopt Lite-Mono[54] as our preferred disparity estimator, achieving an optimal balance between training efficiency and reconstruction quality.

A.7 Details of the real-world multi-device ZI test sets

To comprehensively evaluate the generalization ability of different ZI methods, we construct real-world multi-device test sets spanning four mainstream smartphone platforms: HuaweiPura70Ultra, RedmiK50Ultra, OPPOFindX7Ultra, and iPhone16ProMax. For each device, we collect a substantial number of dual-camera zoom sequences under diverse environmental conditions and usage scenarios.

The complete test sets consist of 499 sequences, including 160 from Huawei, 69 from OPPO, 170 from iPhone, and 100 from Redmi. These sequences span a broad range of real-world scenes, such as indoor environments (e.g., offices, shopping malls, cafés) and outdoor locations (e.g., parks, sidewalks, playgrounds), with variations in illumination, motion intensity, and zoom transitions to reflect practical deployment conditions.

Image resolution varies by device: Huawei (2580×1560), Redmi (1216×1632), iPhone (2016×1512), and OPPO (2048×1536). Each sequence captures paired wide and main camera images along the zoom trajectory, providing a challenging and diverse benchmark for evaluating ZI methods. Representative samples from the test sets are shown in Figure 10.

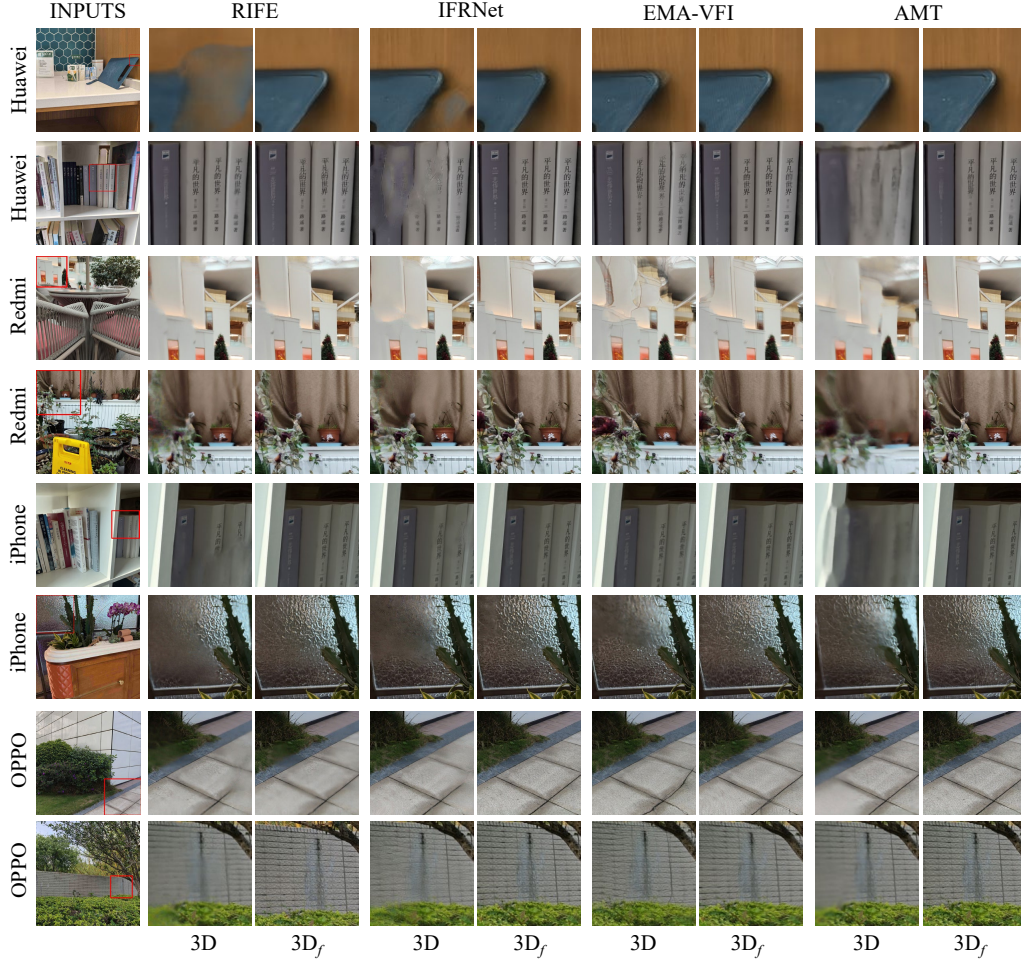


Figure 11: Additional visual comparisons on real-world ZI test data from four smartphone platforms. Subscript f indicates models finetuned on our ZI dataset. Finetuned models produce sharper structures and improved geometric consistency across diverse scenes.

A.8 Additional visual results of OmniZoom on the real-world ZI test sets

To further evaluate the generalization capability of our method, we present additional visual comparisons on the real-world ZI test sets. Specifically, we compare 3D-TPR models before and after finetuning on the ZI dataset across four smartphone platforms. As shown in Figure 11, the finetuned model produces more temporally consistent and perceptually sharper interpolations, particularly in challenging regions involving parallax, fine textures, or low-light conditions. These results further demonstrate the effectiveness of our ZI dataset in enhancing cross-device generalization for real-world zoom interpolation.

A.9 Benchmarking ZI dataset across 1D, 2D, and 3D-TPR frameworks

The 1D-indexed models are directly adopted from publicly released versions of each corresponding FI network. The 2D-indexed counterparts follow the interpolation strategy introduced in [60]. For each framework, we finetune the models using our proposed ZI dataset and evaluate their performance on the real-world multi-device test sets. Both qualitative and quantitative comparisons are conducted to assess the improvements enabled by ZI supervision under different indexing paradigms.

Quantitative comparisons. We conduct a comprehensive evaluation of the 1D, 2D, and 3D-TPR indexing frameworks across four FI models and four real-world smartphone platforms, considering both the base models and those finetuned on our ZI dataset. As shown in Table 9, the 3D-TPR framework

Table 9: Quantitative result of 1D, 2D, and 3D-TPR frameworks across four FI models and four devices. All models are evaluated before and after finetuning on our ZI dataset. **Bold** indicates the best performance after finetuning, and underline marks the best result before finetuning. 3D-TPR framework reliably achieves superior perceptual quality, and its finetuned variant ($3D_f$) outperforms others in most cases, demonstrating the effectiveness of our ZI dataset and the 3D-TPR design.

Device	Metrics	RIFE						IFRNet					
		1D	2D	3D	$1D_f$	$2D_f$	$3D_f$	1D	2D	3D	$1D_f$	$2D_f$	$3D_f$
Huawei	NIQE↓	3.9181	<u>3.8464</u>	3.8651	3.8678	3.8170	3.7422	3.6995	3.6801	<u>3.4852</u>	3.6044	3.6455	3.5035
	PI↓	4.2567	4.2505	<u>4.1537</u>	4.0698	4.0599	3.9360	4.0141	3.9570	<u>3.3150</u>	3.6448	3.6136	3.2520
	CLIP-IQA↑	0.4078	0.3691	<u>0.4939</u>	0.4573	0.4858	0.5233	0.4910	0.5422	<u>0.5784</u>	0.5649	0.5770	0.5909
	MUSIQ↑	49.4314	44.8268	<u>58.8362</u>	58.4623	58.7098	60.8585	51.2033	57.4632	<u>73.0233</u>	61.2380	63.1001	74.1220
Redmi	NIQE↓	5.2253	5.0138	<u>4.8764</u>	4.6925	4.5118	4.4720	5.3165	5.0098	<u>4.2837</u>	4.6223	4.5292	4.0695
	PI↓	5.6387	5.3615	<u>5.0247</u>	4.8289	4.6331	4.5195	5.5147	5.1108	<u>3.5993</u>	4.6454	4.5053	3.3880
	CLIP-IQA↑	0.3661	0.4077	<u>0.4664</u>	0.4559	0.4851	0.4930	0.3691	0.4219	<u>0.4913</u>	0.4765	0.4947	0.5152
	MUSIQ↑	51.7491	56.3870	<u>61.9235</u>	60.6873	62.0812	63.6990	51.3791	57.5453	<u>73.7573</u>	61.7805	62.4252	74.3871
iPhone	NIQE↓	4.3433	4.2031	<u>4.1027</u>	3.9339	3.9532	3.8601	4.0136	3.8786	<u>3.6923</u>	3.7891	3.8169	3.6953
	PI↓	4.6718	4.5340	<u>4.3503</u>	4.1674	4.2135	4.0657	4.4137	4.2090	<u>3.3942</u>	4.0014	3.9636	3.3165
	CLIP-IQA↑	0.4476	0.4821	<u>0.5363</u>	0.5371	0.5345	0.5492	0.4814	0.5240	<u>0.5829</u>	0.5666	0.5778	0.5949
	MUSIQ↑	52.2715	55.0577	<u>59.6800</u>	59.2367	58.8978	60.7988	52.7781	57.4088	<u>73.3569</u>	60.8860	61.9734	73.8292
OPPO	NIQE↓	5.2103	4.6364	<u>4.6100</u>	5.0142	4.5575	4.3968	5.9164	4.5820	5.2039	5.6720	5.7078	5.1133
	PI↓	5.4442	5.0848	<u>5.0104</u>	5.0965	5.0650	4.6815	5.1909	4.9749	<u>4.9603</u>	5.0287	4.8435	4.7144
	CLIP-IQA↑	0.4270	0.4989	<u>0.5366</u>	0.5450	0.5699	0.5830	0.5181	<u>0.5525</u>	0.5212	0.5207	0.5667	0.5461
	MUSIQ↑	54.8990	63.2645	<u>65.7912</u>	63.3583	65.0204	67.1494	54.8757	67.2635	<u>75.2718</u>	64.9371	67.2739	75.4545
Device	Metrics	EMA-VFI						AMT					
		1D	2D	3D	$1D_f$	$2D_f$	$3D_f$	1D	2D	3D	$1D_f$	$2D_f$	$3D_f$
Huawei	NIQE↓	4.6877	<u>3.6363</u>	3.8470	3.7737	3.4852	3.8341	4.7231	<u>3.5665</u>	3.6459	3.9866	3.6246	3.4547
	PI↓	3.7060	<u>3.7240</u>	3.8566	3.6638	3.7186	3.8467	4.9499	<u>3.5147</u>	3.6132	3.9974	3.4639	3.3121
	CLIP-IQA↑	0.5619	0.5612	<u>0.5621</u>	0.5588	0.5535	0.5684	0.3127	0.5428	<u>0.5498</u>	0.5898	0.5944	0.6018
	MUSIQ↑	60.4395	61.7263	<u>62.7403</u>	62.8008	63.0230	63.5048	53.6305	71.4369	<u>71.7016</u>	73.4803	72.7456	73.7092
Redmi	NIQE↓	4.5060	4.4088	<u>4.3424</u>	4.4864	4.3537	3.8852	5.9112	5.1925	<u>4.7426</u>	4.9700	5.1082	3.9835
	PI↓	4.6410	4.6708	<u>4.5974</u>	4.5829	4.5665	4.5519	5.0623	5.4335	<u>4.0235</u>	4.8863	5.0036	3.3548
	CLIP-IQA↑	0.4651	0.4599	<u>0.4807</u>	0.4809	0.4812	0.4947	0.4191	0.4336	<u>0.4754</u>	0.5082	0.4944	0.5383
	MUSIQ↑	55.3260	57.0371	<u>59.6610</u>	60.2753	59.9715	60.7579	49.2505	57.6671	<u>71.9982</u>	63.6003	61.2496	74.3325
iPhone	NIQE↓	4.9410	4.8155	<u>4.0555</u>	4.7661	4.2139	4.0934	5.0436	<u>3.5932</u>	3.7053	3.5792	3.5883	3.5087
	PI↓	5.0171	<u>3.9994</u>	4.1304	4.5345	3.8462	4.1123	5.0410	<u>3.4293</u>	3.5367	3.3379	3.3233	3.2661
	CLIP-IQA↑	0.4603	0.5352	<u>0.5387</u>	0.5367	0.5462	0.5503	0.3723	<u>0.5931</u>	0.5930	0.6064	0.6120	0.6211
	MUSIQ↑	55.5380	58.1422	<u>58.6204</u>	57.6496	59.1077	60.1002	57.8120	71.9805	<u>72.1243</u>	73.5609	72.9853	73.6150
OPPO	NIQE↓	5.0104	4.5568	<u>4.5166</u>	4.9704	4.5293	4.5071	5.7811	<u>4.9042</u>	5.3017	4.7389	4.7885	4.6699
	PI↓	5.1828	4.9569	<u>4.9022</u>	5.0729	4.9006	4.8752	5.8036	<u>4.9676</u>	5.3808	5.0513	4.9911	4.5702
	CLIP-IQA↑	0.5820	<u>0.5622</u>	0.5590	0.5071	0.5502	0.5591	0.3499	<u>0.4798</u>	0.4518	0.4673	0.5461	0.5595
	MUSIQ↑	67.0395	67.4327	<u>67.5921</u>	67.1736	67.5282	67.6444	54.9076	<u>73.0842</u>	72.1563	72.8543	74.7868	75.2187

consistently outperforms its 1D and 2D counterparts after finetuning, achieving the best performance (highlighted in bold) across nearly all perceptual quality metrics. Notably, even without finetuning, 3D-TPR frequently ranks second (underlined), demonstrating strong generalization capability and architectural robustness. In contrast, 1D and 2D variants exhibit only limited performance gains, particularly under challenging cross-device settings. These results support two conclusions: (1) while our ZI dataset provides performance gains across all indexing paradigms, the improvements are most pronounced under the trajectory-aware 3D-TPR framework; and (2) OmniZoom (i.e., $3D_f$ -TPR) offers a universal, plug-and-play solution for real-world cross-device zoom interpolation, effectively bridging domain gaps and enhancing perceptual quality across diverse hardware platforms.

Qualitative comparisons. Beyond quantitative evaluation, we provide qualitative results in Figures 12 and 13, illustrating ZI outputs on Huawei, Redmi, iPhone, and OPPO devices across four FI networks. Each figure presents six variants per network: 1D, 2D, and 3D, along with their finetuned counterparts ($1D_f$, $2D_f$, and $3D_f$).

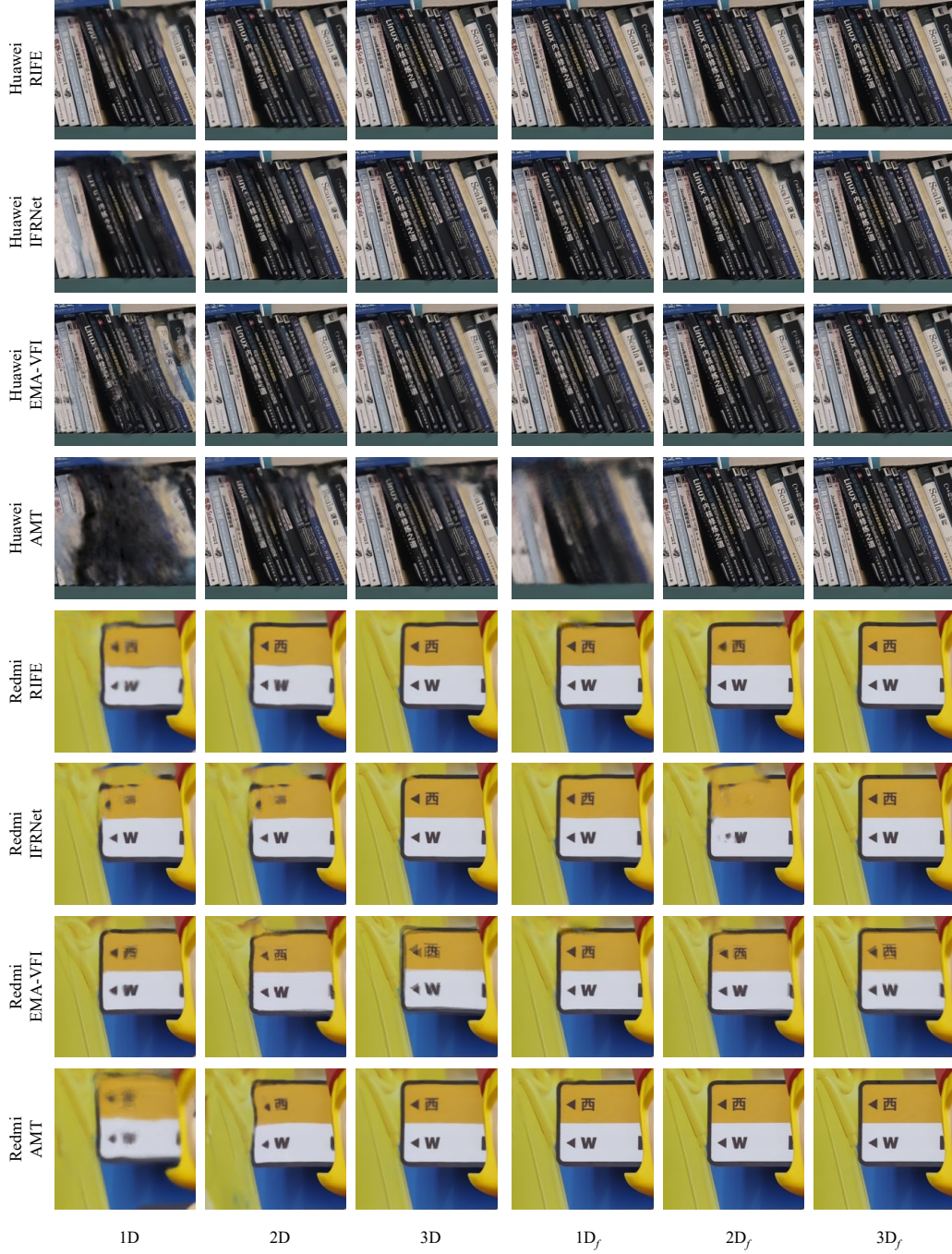


Figure 12: Qualitative comparisons on Huawei (top four rows) and Redmi (bottom four rows). While 1D and 2D models exhibit noticeable blurring and detail loss across networks, $3D_f$ effectively restores fine details, producing sharper structures and improved local contrast.

Several key observations emerge from these visualizations. First, the 3D configuration consistently produces sharper and more coherent results than its 1D and 2D counterparts, highlighting the effectiveness of the 3D-TPR framework in real-world ZI tasks. Second, across all indexing paradigms, all finetuned models show clear perceptual improvements, demonstrating the broad applicability of our ZI dataset across diverse FI architectures.

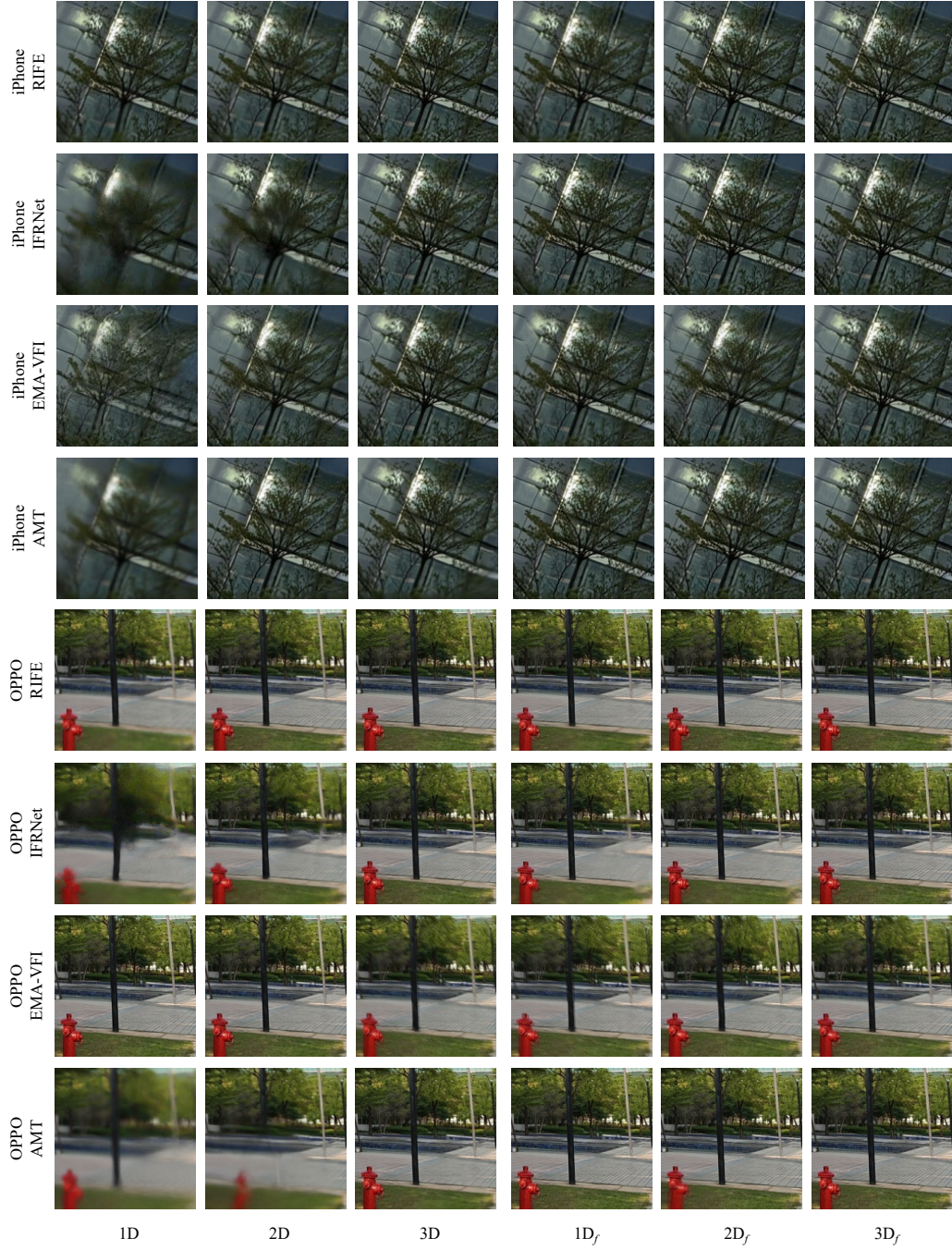


Figure 13: Qualitative comparisons on iPhone and OPPO test sets under rich structural conditions. While 1D and 2D variants exhibit edge instability and perceptual artifacts, $3D_f$ produces noticeably clearer results with sharper contours and reduced blurring, effectively preserving structural fidelity.

Notably, $3D_f$ achieves the highest perceptual quality in nearly all cases, with fewer artifacts and enhanced texture fidelity. For example, on Redmi sequences, $3D_f$ recovers sharper character contours and fine line structures compared to other variants. On iPhone and OPPO, where scenes contain more intricate textures and geometric structures, 1D and 2D variants frequently introduce ghosting and edge artifacts, issues that are substantially mitigated by the $3D_f$ configuration.

These results underscore the strength of OmniZoom as a unified, cross-device solution for plug-and-play zoom interpolation, offering robust perceptual quality across various hardware platforms.

Table 10: Upper bound evaluation of the proposed 3D-TPR framework on four FI networks using ground-truth 3D supervision. The 2D framework serves as a baseline [60]. All experiments on 3D use the 3D trajectory progress ratio as timestep map. 3D-t-m indicates using only 3D-TPR encoding; 3D-m adds texture-focus strategy; 3D-t includes mask penalty constraint; and 3D integrates all components. **Bold** indicates improvements over 2D baseline.

Network	RIFE			IFRNet			EMA-VFI			AMT		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
2D	27.4050	0.9010	0.0863	27.1276	0.8994	0.0780	24.7313	0.8514	0.0810	27.1726	0.9017	0.0807
3D-t-m	28.7510	0.9267	0.0773	28.2484	0.9225	0.0715	25.4078	0.8618	0.0780	28.4850	0.9269	0.0837
3D-m	28.7311	0.9264	0.0764	28.3486	0.9241	0.0690	25.4705	0.8631	0.0800	28.5942	0.9285	0.0767
3D-t	28.5518	0.9234	0.0764	28.1917	0.9206	0.0686	25.3557	0.8618	0.0781	28.5895	0.9277	0.0810
3D	28.6548	0.9255	0.0773	28.2967	0.9220	0.0656	25.4984	0.8633	0.0770	28.6271	0.9286	0.0768

A.10 Upper bound analysis of 3D-TPR framework

To evaluate the performance ceiling of the 3D-TPR framework, we adopt ground-truth 3D geometry as the timestep map, thereby ensuring perfect consistency between training and inference. As shown in Table 10, we conduct a complementary ablation study under this setting for fair comparison.

Results indicate that utilizing consistent 3D similarity maps during both training and testing leads to substantial gains across all metrics and networks: PSNR improves by over 1.0 dB on average, SSIM consistently increases, and LPIPS is significantly reduced. These results reveal the latent capacity of the 3D-TPR framework when accurate geometry is available throughout.

This upper bound analysis provides valuable insight into the theoretical capacity of our method. While full 3D supervision may not be available in practice, our empirical findings suggest that leveraging lightweight geometry estimation modules or substituting with physically obtainable priors (e.g., depth from stereo, SLAM-based pose, or monocular depth) can approximate similar benefits. This observation offers practical guidance for future extensions that aim to balance reconstruction quality and computational feasibility.

A.11 Additional visual results of 3D-TPR framework

We present additional qualitative comparisons between the proposed 3D-TPR framework and conventional 2D baselines, as shown in Figure 14. The results demonstrate that 3D-TPR effectively restores degraded regions in 2D-based outputs, particularly in areas suffering from motion-induced blur where structural details are severely corrupted or entirely missing.

Moreover, 3D-TPR alleviates misalignment artifacts commonly encountered in 2D models when handling textures with high levels of repetition and self-similarity, such as linear structures or dot-like patterns, by incorporating trajectory-aware spatial priors. In addition to these localized corrections, our method consistently produces globally improved visual quality, exhibiting sharper details, better structural coherence, and a more realistic overall appearance. These findings further highlight the advantages of integrating 3D spatial awareness into temporal interpolation tasks.

A.12 Broader impacts

This work is motivated by the goal of enhancing frame interpolation in zoom scenarios, with direct benefits for a wide range of applications in mobile photography, video conferencing, augmented reality (AR), and digital content creation. Our proposed framework improves the temporal smoothness and spatial consistency of interpolated video frames across heterogeneous smartphone platforms, contributing to better visual quality in real-world capture workflows. In particular, it enables smoother zoom transitions without requiring any hardware-level changes due to zero cost, offering cost-effective enhancements for existing consumer devices.

Such capabilities are especially impactful in mobile videography, where users often experience abrupt visual discontinuities during zoom. Our method can help reduce motion artifacts and maintain semantic coherence in challenging scenes, which is beneficial for video-based communication,

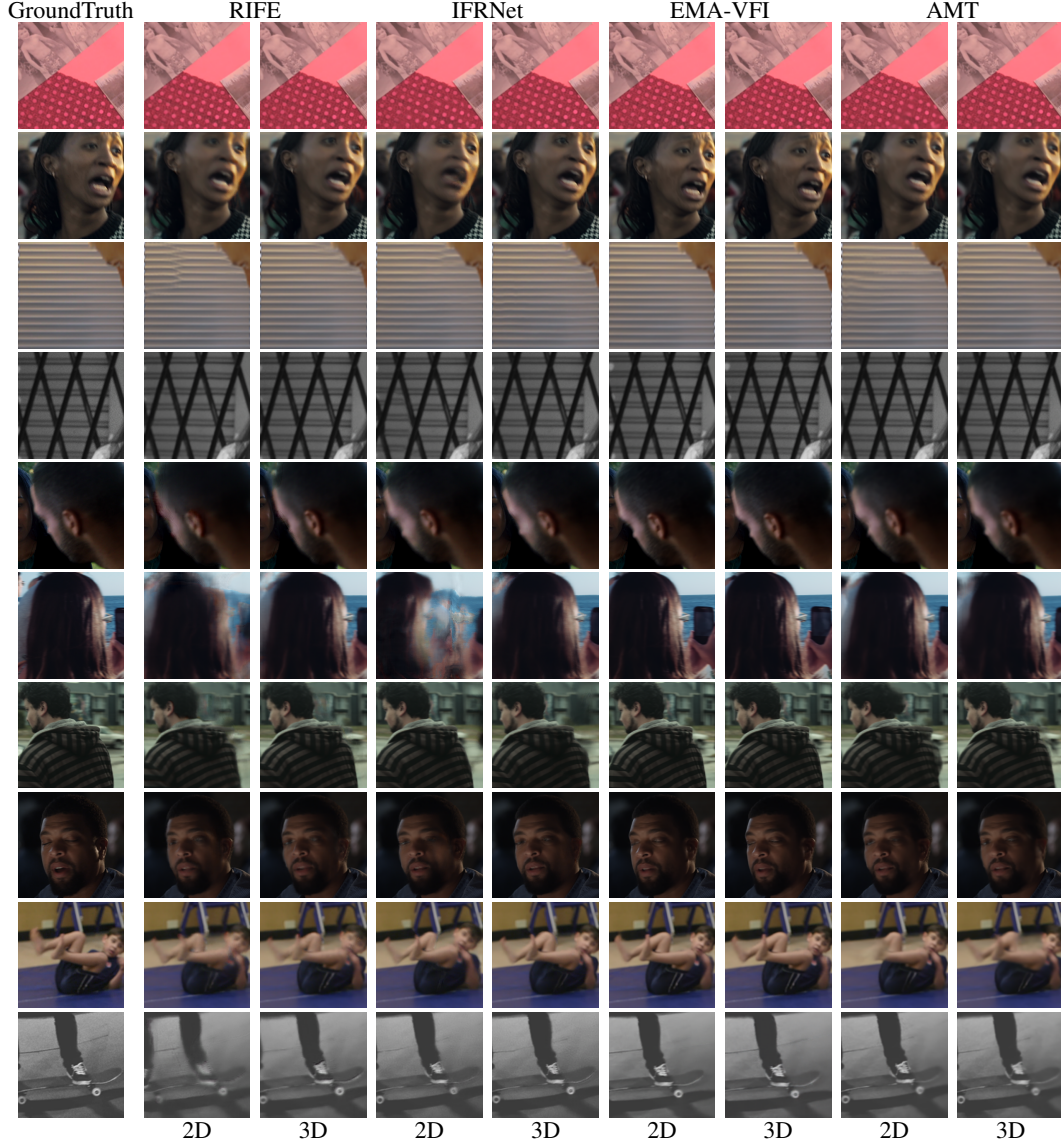


Figure 14: Visual comparison of 2D and 3D interpolation results across networks at the same timestep. The 3D method consistently yields less-blurred and neater outputs across all networks.

content sharing on social platforms, and immersive AR experiences that rely on high frame fidelity. Furthermore, cross-device compatibility and plug-and-play integration facilitate broader adoption across hardware vendors, making advanced interpolation accessible without retraining for each model.

Nevertheless, as with other generative video technologies, improvements in visual realism may pose risks if misused, such as enabling the creation of misleading or manipulated video content. While our framework is restricted to supervised frame interpolation under dual-camera settings and does not support arbitrary video generation, we emphasize the importance of responsible use. To mitigate misuse, we do not release general-purpose generative models, and our dataset is limited to constrained zoom interpolation scenarios. We encourage future efforts to pair such technologies with appropriate safeguards and transparency mechanisms to ensure ethical deployment.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: As stated in both the abstract and the introduction, our contributions and scope are clearly defined.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are discussed in the Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The full set of assumptions and derivations is provided in Section 3 and Appendix A.4.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide detailed descriptions of the ZI dataset construction and 3D-TPR architecture in the section3 and Appendix A.1 and Appendix A.4. In addition, the Section 4 and Appendix A.3 include specific training and evaluation settings. These details are sufficient to reproduce the core experimental results and verify our main conclusions.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We intend to release the code and dataset upon publication. While they are not publicly available at the time of submission, the release is scheduled and will include detailed documentation to ensure full reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All essential training and testing details are described in Section 4, Appendix A.1, and Appendix A.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: While we do not include error bars or formal statistical significance tests, we report average metrics (e.g., PSNR, SSIM, LPIPS) over all test samples to ensure statistical reliability of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the compute resources used for all experiments in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: We have thoroughly reviewed the NeurIPS Code of Ethics and ensured that our research adheres to all stated ethical guidelines.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both potential positive societal impacts and negative societal impacts of the work performed in Appendix A.12

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not involve pretrained language models, image generation, or internet-scraped datasets, and therefore poses no apparent risk of misuse or dual-use.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All third-party datasets and models used in our paper are clearly cited with their original publications.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: We introduce a new cross-device ZI dataset in this work. While it is not publicly available at submission time, we plan to release it upon publication. The dataset construction process is thoroughly described in Section 3.1, Section 4, Appendix A.1, and Appendix A.2, and full documentation will be provided alongside the release.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve any crowdsourcing experiments or human subject research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This work does not involve any human subjects or crowdsourced data collection, so no IRB approval was required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were only used for language polishing and did not contribute to the core methodology, experimental design, or scientific content of the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.